



UNIVERSIDADE DO ESTADO DO PARÁ – UEPA
CENTRO DE CIÊNCIAS NATURAIS E TECNOLOGIA – CCNT
BACHARELADO EM ENGENHARIA DE SOFTWARE

Alefe Filipe
Caio Martins
Ismael Henrique
Djunio Filho

Documentação: Carona um sistema web para mobilidade universitária

Redenção
2020.2

Sumário

1 - Introdução.....	3
2 - Objetivo.....	3
3 - Justificativa.....	3
4 - Modelo de Processo de software em Cascata.....	4
4.1-Motivação.....	4
4.2- Vantagens	4
4.3- Desvantagens.....	4
5 - Mapeamento do modelo de processo de software em cascata.....	5
6 - Análise e definição de requisitos.....	5
6.1- Levantamento de requisitos.....	5
6.2- Requisitos funcionais.....	5
6.3- Requisitos não funcionais.....	5
6.4- Requisitos não funcionais organizacionais.....	5
6.5- Requisitos não funcionais externos.....	6
7 - Modelagem de dados.....	7
8 - Diagrama de casos de uso.....	8
9 - Projeto de sistema de software.....	8
10 - Implementação e teste de unidade.....	9
10.1- Plano de teste.....	9
10.2- Casos de teste.....	8
10.3- Relatório de erros.....	9
11 - Integração e teste de sistema.....	11
12 - Referências bibliográficas.....	12

1 - INTRODUÇÃO

Neste Documento vamos versar sobre a adversidade que tem afetado muitos integrantes da comunidade UEPA, que é sua locomoção pela cidade, seja para transitar até a universidade, ou para seus afazeres individuais. Pretendemos por meio dessa plataforma oferecer serviços de diversas formas, a título de exemplo: Caronas por meio de carro, van, ônibus, moto e bicicleta. Além dessas formas tradicionais de oferecer transporte, caso deseje provemos uma companhia para uma caminhada ou passeio.

Tendo ciência desse impedimento, desenvolvemos uma plataforma web que facilite o transporte até o local desejado ou até mesmo apenas uma companhia para seus compromissos.

O projeto foi colocado em prática através do modelo de processo de software em cascata, nele detalhamos todo o processo desde a proposta do sistema até a instalação e manutenção de desenvolvimento da plataforma. Além de utilizar o modelo de processo de software em cascata para criar o projeto, utilizamos a ferramenta de gerência "Trello". Trello é uma ferramenta de colaboração que organiza seus projetos em quadros. De relance, Trello informa o que está sendo trabalhado, quem está trabalhando em quê, e onde algo está em um processo.

2 - OBJETIVO

Desenvolver uma aplicação web para estudantes, professores e colaboradores da UEPA que permita realizar um cadastro e por meio de suas informações solicitar e disponibilizar caronas a fim de promover a locomoção dentro da cidade e melhorar a comunicação entre os mesmos.

3 - JUSTIFICATIVA

Devido a alguns funcionários, professores e alunos não possuírem condução, partindo dessa premissa houvesse a necessidade de se pensar em uma solução que poderia estar amparando de alguma forma a locomoção dessas pessoas dentro da cidade ou a caminho da própria universidade. Conforme municiamos na tabela abaixo:

O problema de:	Locomoção dos discentes, docentes e colaboradores da UEPA de forma gratuita;
Afeta:	Discentes, docentes e colaboradores;
Cujo impacto é:	Falta de transporte público e segurança na cidade;
Uma boa solução seria:	Desenvolver uma solução web que seja colaborativa junto ao corpo discente e docente para solicitação e disponibilização de caronas dentro da cidade.

4 - MODELO DE PROCESSO DE SOFTWARE EM CASCATA

4.1 - MOTIVAÇÃO

O modelo em cascata considera as atividades fundamentais do processo, compreendendo especificação, desenvolvimento, validação e evolução, e as representa como fases de processos separadas e bem definidas, como por exemplo: análise e definição de requisitos, projeto de software, implementação e teste. O objetivo principal deste sistema é que as diferentes fases de desenvolvimento seguem uma sequência: A primeira etapa se direciona para a segunda e esta se movimenta para a terceira e assim por diante.

4.2 - VANTAGENS

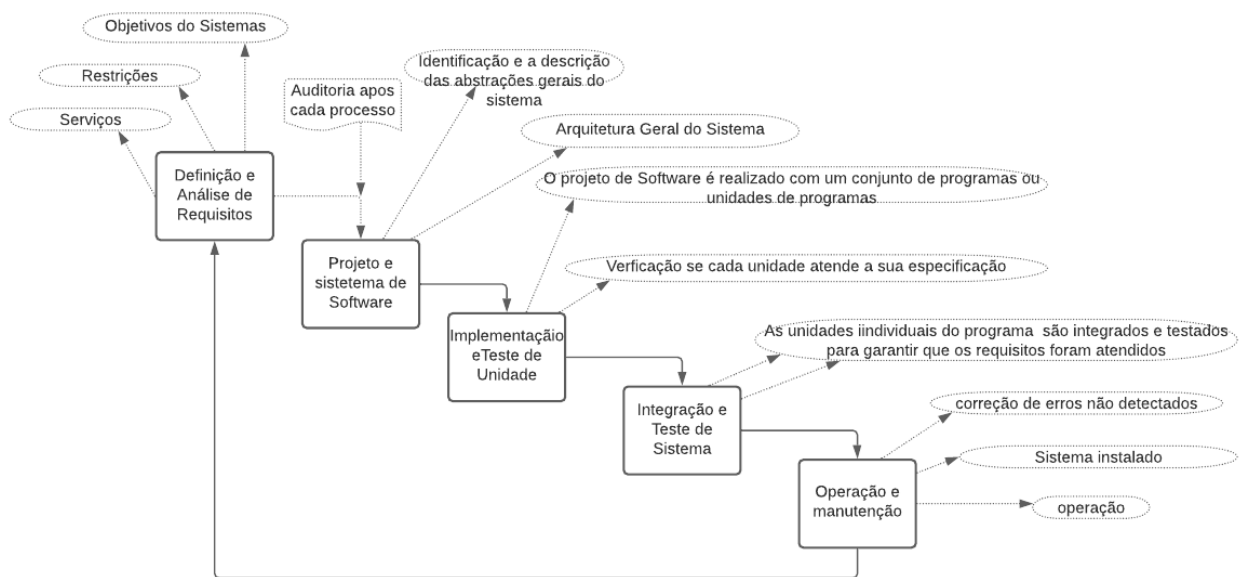
- Documentação sólida em cada processo(completa);
- Aderência a outros modelos;
- Processos bem definidos;
- Pode ser combinado com outro modelo.

4.3 - DESVANTAGENS

- Fases inflexíveis;
- Usar apenas quando os processos de requisitos forem bem definidos e compreendidos;
- Difícil se adequar às mudanças inevitáveis;
- Uma versão usável do software só ficará pronta no final.

5 - MAPEAMENTO DO MODELO DE PROCESSO DE SOFTWARE EM CASCATA

figura 1- cascata



6 - ANÁLISE E DEFINIÇÃO DE REQUISITOS

6.1 - Levantamento de requisitos

O levantamento de requisitos é um processo no qual serve para absorver e documentar as necessidades do cliente, antes de planejar o desenvolvimento do software, levando em consideração que os problemas solucionados pela aplicação, serão problemas reais. Os requisitos levantados podem ser divididos em dois tipos, os **requisitos funcionais** (RF) e os **requisitos não funcionais** (RNF).

6.2 - Requisitos funcionais

São os requisitos relacionados ao uso de software em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas. Essas qualidades dizem respeito a como as funcionalidades serão entregues ao usuário do programa. Descrevem o comportamento do sistema e seus requisitos para a execução de cada item, um requisito é uma propriedade que um software exibe para solucionar possíveis problemas reais, quando se trata de um software sob demanda, por exemplo:

- [RF 001]: O sistema deve permitir que o usuário se registre;
- [RF 002]: O sistema deve autorizar que o usuário faça login;
- [RF 003]: O sistema não deve permitir que mais de um usuário se cadastre com o mesmo e-mail;
- [RF 004]: O usuário poderá visualizar, atualizar ou excluir seus dados;
- [RF 005]: O usuário deverá estar matriculado na UEPA;
- [RF 006]: Caso o usuário priorize uma carona do sexo feminino, deverá ser atendido;
- [RF 007]: O sistema deve oferecer as opções de carona a pé(companhia), moto, carro, ônibus, van e bicicleta.

6.3 - Requisitos não funcionais

Os requisitos não funcionais são aqueles que não possuem relação com a funcionalidade do software, são considerados como atributos de qualidades, e tem grande importância na tomada de decisão no decorrer do desenvolvimento do software.

6.4 - Requisitos Não funcionais Organizacionais

- [RNF 001]: **implementação**: O sistema deverá ser desenvolvido na linguagem de programação: *Javascript e PHP 2.0*-Linguagem: *Html e css 43.0*-Biblioteca: *bootstrap*;
- [RNF 002]: **Entrega**: 07/12/2021

6.5 - Requisitos Não funcionais Externos

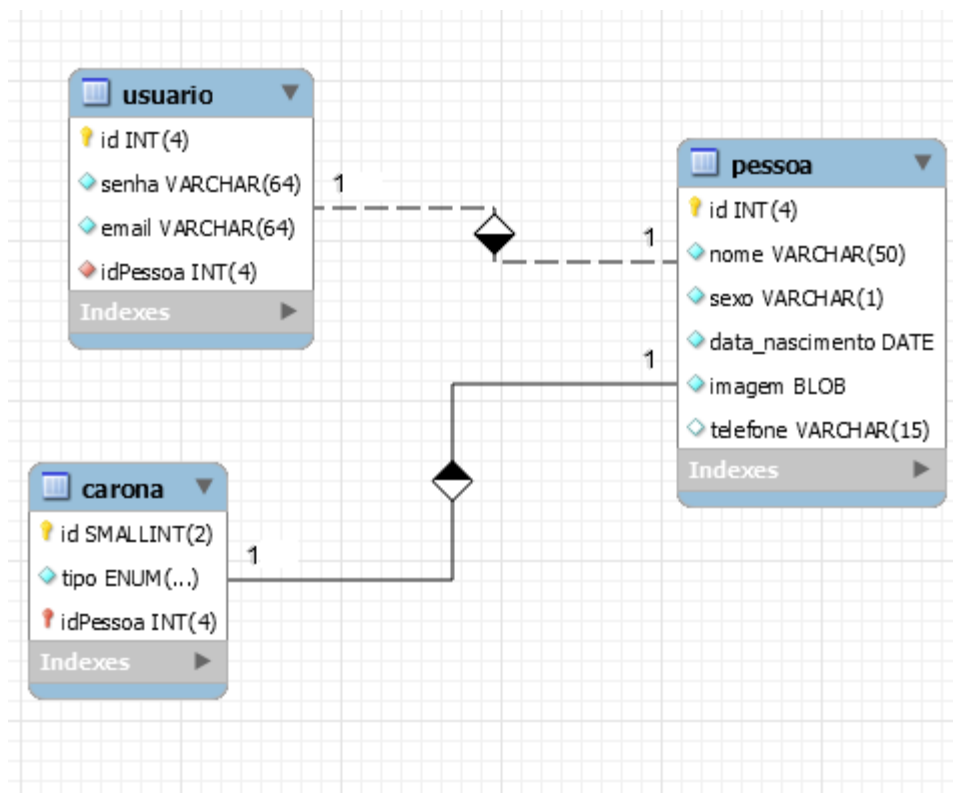
- [RNF 003]: **Ético**: Nenhum dados serão tratados sem a autorização do usuário
- [RNF 004]: **legais**: O sistema deverá atender às normas legais, tais como padrões e leis, tal como: **lei nº 13.709, de 14 de agosto de 2018**(Lei Geral de Proteção de Dados) e **lei nº 12.965, de 23 de abril de 2014**(Estabelece princípios, garantias, direitos e deveres para o uso da Internet no Brasil.)

- [RNF 005]: **interoperabilidade**: O sistema deverá se comunicar com o banco de dados MySQL.
- [RNF 006] **segurança**: Um usuário não poderá ter acesso aos dados de outro usuário;
- [RNF 007]: **Privacidade**: O sistema não deve manipular, quaisquer informações pessoais sobre os usuários, com exceção do nome, e-mail, nº de matrícula e sexo.

7 - Modelagem de dados

A modelagem de dados tem como objetivo explorar estruturas orientadas a dados, pode ser utilizado para uma variedade de propósitos, desde modelos conceituais até modelos físicos, partindo de um ponto de vista de um desenvolvedor orientado a objetos, a modelagem de dados busca especificar as classes e seus atributos. Em nosso projeto buscamos modelar o banco de dados, suas entidades, atributos e relacionamentos.

Figura – 2: modelagem dos dados

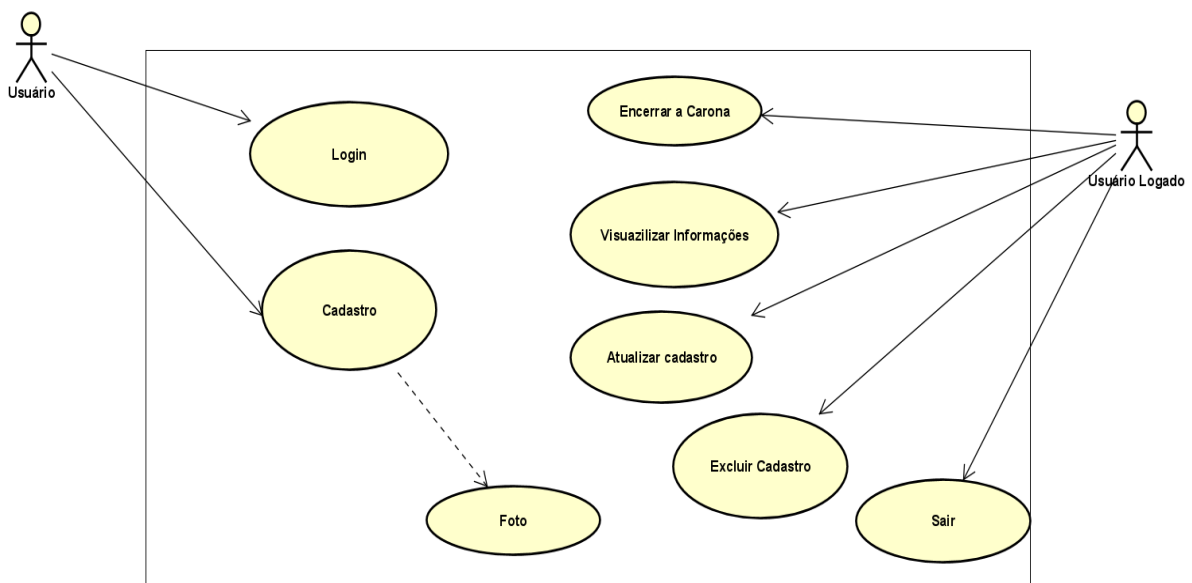


As ligações entre as tabelas acima, significa relacionamento entre elas, a notação 1 - 1, significa que uma pessoa pode disponibilizar uma e somente uma carona, assim como uma pessoa pode ter um e apenas um usuário.

8 - DIAGRAMA DE CASO DE USO

Diagrama de caso de uso tem como objetivo demonstrar como será a interação do usuário em relação ao sistema, como se dispõe descrito na Figura 3.

Figura – 3: diagrama de caso de uso



9 - PROJETO DE SISTEMA E SOFTWARE

O processo de sistema divide os requisitos em sistemas de hardware ou de software. Ele estabelece uma arquitetura geral do que é necessário para o sistema operar.

9.1 - Requisitos de software:

XAMPP VERSÃO 5.4.27 / PHP Composer versão 1.2.2 ou superior NetBens para php Navegador web Google Chrome na versão 58.0.3029.110 ou superior, Mozilla Firefox versão 53.0.3 ou superior.

Sistema Operacional: Windows XP, 7, 8, 10, ou 11

Sistemas Mobile: Android ou IOS

9.2 - Requisitos de Hardware:

Processador: 1,9 giga-hertz (GHz) ou um processador dual core de x86 ou x64 bits com conjunto de instruções SSE2

Memória: 2 GB de RAM

Exibição: Super VGA com resolução de 1024 x 768

10 - IMPLEMENTAÇÃO E TESTE DE UNIDADE

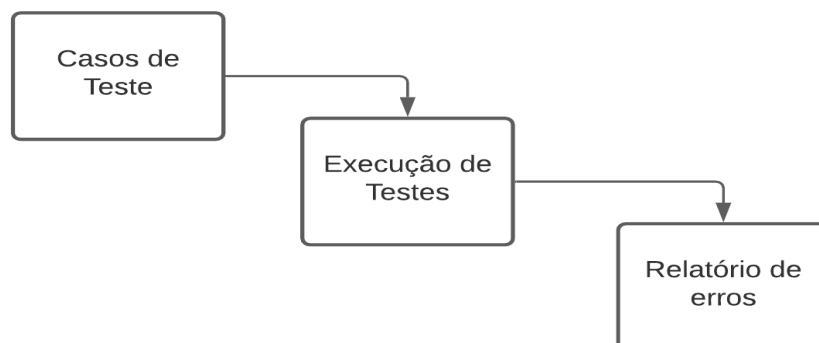
Nesse estágio, o projeto de software é realizado como um conjunto de programas ou unidades de programa. O teste unitário envolve a verificação de que cada unidade atende à sua especificação de requisitos.

10.1 - Plano de teste

O plano de teste possui duas metas:

- Demonstrar ao desenvolvedor e ao cliente que o software atende aos requisitos, isso significa que deve haver pelo menos um teste para cada requisito.
- Descobrir falhas ou defeitos no software que apresenta comportamento incorreto, ou em não conformidade com a sua especificação. O teste de defeitos está relacionado à remoção de todos os tipos de comportamentos indesejáveis do sistema, como travamentos, interações indesejáveis, cálculos incorretos e corrompimento de dados.

figura 4- plano de teste



10.2 - Casos de teste

1º teste: cadastrar um usuário;


2º teste: testar o login do usuário;

3º teste: testar a atualização e exclusão dos dados;

4º teste: o pedido da carona.

10.3 - Relatório de erros

1º teste: O cadastro funcionou com eficiência.

**Uber Universitário**

Registro

Nome:

Email:

Telefone:

Senha:

Confirme sua senha:

Data de Nascimento:

Sexo:

☐ Masculino

☐ Feminino

Adicionar foto:

Nenhum arquivo escolhido

Você oferece algum tipo de carona?

Termos de uso:

Política de Privacidade:

...OS TERMOS E CONDIÇÕES É DE EXTREMA IMPORTÂNCIA QUE SEJAM LIDOS, NELE LISTAMOS PORQUE ESTAREMOS RECOLHENDO TAIS INFORMAÇÕES, QUAIS INFORMAÇÕES, E COMO FAREMOS O

☐ Aceito

2º teste: Login do usuário:

Entrar no sistema

Email


Senha

3º teste: Atualização e exclusão de dados:


Matrícula	Nome	Email	Sexo	Data de Nascimento	Ação
20215550042	Ismael	ismael.silva@aluno.uepa.br	M	05/12/2000	Info Atualizar Excluir

[Sair](#) [Deseja encerrar o serviço de carona?](#) [Pesquisar](#)

4º teste: Pedido da carona:



Uber Universitário

Foto	Matrícula	Nome	Sexo	Tipo de Carona	Ação
	20215550011	alefe	M	VAN	Solicitar

[Voltar](#)

11 - INTEGRAÇÃO E TESTE DE SISTEMA

As unidades individuais do programa são integradas e testadas como um sistema completo para garantir que os requisitos de software foram atendidos. Fizemos os testes do Carona, tela por tela e os erros encontrados foram retificados, os requisitos propostos pelo sistema foram atendidos, sem grandes falhas em cada fase. O resultado atual é mais que satisfatório.

12 - REFERÊNCIAS BIBLIOGRÁFICAS:

SOMMERVILLE, IAN. **Engenharia de Software**. ed.8. São Paulo, Pearson, 2010.

