

*Escola Superior de Tecnologia e Gestão    
Instituto Politécnico da Guarda*

PLANEAR A PRODUÇÃO COM BASE NAS ENCOMENDAS

RUI CONDESSO 1701429

RICARDO SOUSA 1705109

PEDRO MATOS 1700789

ENGENHARIA DE SOFTWARE II

Dezembro de 2022

Índice de Conteúdos

[1. Introdução 3](#_Toc124118300)

[2. Estado Arte 4](#_Toc124118301)

[1. MRpeasy 4](#_Toc124118302)

[2. OpenProject 4](#_Toc124118303)

[3. Diagrama de Casos de Uso 6](#_Toc124118304)

[3.2 Gerir previsões da entrega das encomendas 9](#_Toc124118305)

[3.3 Diagrama de Sequencia 10](#_Toc124118307)

[4. User Stories 11](#_Toc124118308)

[5. Diagrama de Classes 13](#_Toc124118309)

[6. Diagrama de Estados 17](#_Toc124118310)

[7. Diagrama de Pacotes 18](#_Toc124118311)

[8. Diagramas de Instalação 20](#_Toc124118312)

[9. Tabelas de Casos de Teste 21](#_Toc124118313)

[10. Protótipos 22](#_Toc124118314)

[11. Conclusão 24](#_Toc124118315)

[12. Referencias Biográficas 25](#_Toc124118316)

[13. Anexos 26](#_Toc124118317)

[14. Autoavaliação 29](#_Toc124118318)

Índice de Figuras

[Figura 3.1 Casos de uso 7](#_Toc124118332)

[Figura 2-Diagrama de Sequencias 11](https://ipgpt-my.sharepoint.com/personal/1705109_sal_ipg_pt/Documents/Relatorio%20Engenharia%20de%20software%20II.docx#_Toc124118333)

[Figura 5.1 Diagrama de classes 14](#_Toc124118334)

[Figura 7.1 Diagrama de Pacotes 19](#_Toc124118335)

[Figura - Diagrama de Componentes 20](#_Toc124118336)

[Figura 8.1 Diagrama de instalação 21](#_Toc124118337)

[Figura -Página Inicial do projeto 23](#_Toc124118338)

[Figura -Pagina Login do Projeto 23](#_Toc124118339)

[Figura -Pagina StockFinalProduct 24](#_Toc124118340)

[Figura -Pagina ModelParts 24](#_Toc124118341)

[Figura 1.13.1 Trello do grupo https://trello.com/b/BjVoUXpE/planejar-produ%C3%A7%C3%A3o-com-base-nas-encomendas 27](#_Toc124118342)

[Figura 13.2 Trello do grupo (Cont.) https://trello.com/b/BjVoUXpE/planejar-produ%C3%A7%C3%A3o-com-base-nas-encomendas 27](#_Toc124118343)

[Figura - Trello Final 28](https://ipgpt-my.sharepoint.com/personal/1705109_sal_ipg_pt/Documents/Relatorio%20Engenharia%20de%20software%20II.docx#_Toc124118344)

[Figura -Pagina do GitHub - https://github.com/noellopes/CarManufactoring 29](#_Toc124118345)

Índice de Tabelas

[Tabela 2.2.1 Requisitos funcionais de MRpeasy 4](#_Toc124118268)

[Tabela 2.2.2 Requisitos funcionais de OpenProject 4](#_Toc124118269)

[Tabela 2.2.3 Tabela de funcionalidades 5](#_Toc124118270)

[Tabela 4.2 Descrição de Casos de Uso" Gerir previsões da entrega das encomendas" 9](https://ipgpt-my.sharepoint.com/personal/1705109_sal_ipg_pt/Documents/Relatorio%20Engenharia%20de%20software%20II.docx#_Toc124118271)

[Tabela 4.1 Tabela de User Stories 12](#_Toc124118272)

[Tabela 5.1 Classe Pecas 14](#_Toc124118273)

[Tabela 5.2 Tabela Modelo\_Carro 14](#_Toc124118274)

[Tabela 5.3 Tabela Modelo\_Carro\_Pecas 15](#_Toc124118275)

[Tabela 5.4 Tabela Modelo\_Carro\_Encomenda 15](#_Toc124118276)

[Tabela 5.5 Tabela Operacao inserir na tabela Pecas 16](#_Toc124118277)

[Tabela 5.6 Tabela Operacao inserir na tabela Modelo\_Carro\_Pecas 16](#_Toc124118278)

[Tabela 5.7 Operacao inserir na tabela Modelo\_Carro 16](#_Toc124118279)

[Tabela 6.1 Diagrama de estados 17](#_Toc124118280)

[Tabela 6.2 Classe produção 17](#_Toc124118281)

[Tabela 9.1 Tabela de casos de teste inserir novo modelo e peça na tabela ModelParts 21](#_Toc124118282)

[Tabela 9.2 Tabela de casos de teste inserir novo modelo de Carros na tabela CarModels 21](#_Toc124118283)

# Introdução

No âmbito das cadeiras Engenharia de Software II e Programação para a Internet foi proposto a turma desenvolver um software que tivesse como objetivo a gestão de uma empresa de produção de automóveis.

O desenvolvimento desse mesmo software foi dividido em várias partes sendo que o nosso grupo ficou encarregue da parte do "planeamento da produção com base nas encomendas".

# Estado Arte

1. MRpeasy

Tabela 2.2.1 Requisitos funcionais de MRpeasy

|  |
| --- |
| Requisitos funcionais |
| Planear a produção de forma automática mediante pedido |
| Programar a produção(cronograma) |
| Permitir o reagendamento da produção através de drag and drop no cronograma |
| Fazer a gestão do inventário |
| Mover o stock |
| Permitir o Tracking por números de série |
| Definir níveis ideais de stock e otimização dos já existentes de forma a evitar a falta dos mesmos |
| Manter histórico de operações do stock |
| Calcular custos de produção |
| Calcular o tempo de entrega |
| Ver todo o processo de produção e entrega ao cliente através de uma pipeline |
| Permitir aceder a todos os dados em tempo real através de um pc ou um dispositivo móvel |
| Efetuar previsões de procura e vendas |
| Monitorar a performance do negócio e de toda a empresa em tempo real. |

1. OpenProject

Tabela 2.2.2 Requisitos funcionais de OpenProject

|  |
| --- |
| Requisitos funcionais |
| Permitir a comunicação entre os membros dos projetos e os stakeholders numa única página |
| Gestores de projeto e managers podem ver o progresso do projeto a qualquer altura em qualquer momento e a partir de qualquer lugar. |
| Os atrasos e os avanços têm de ser visíveis em tempo real |
| Toda a gente deve conseguir ver o progresso e os timings de todas as tarefas e os responsáveis pelas mesmas. |
| Os objetivos, os projetos, os processos, as funções e as dashboards devem ser personalizáveis |
| Deve ser possível ver quantas horas cada pessoa trabalhou e que tarefas executou e com base nisso calcular o seu pagamento |
| Deve ser possível todos os membros do projeto colocarem os requisitos para esse mesmo projeto. |

|  |  |  |  |
| --- | --- | --- | --- |
| Funcionalidades | MRpeasy | SYSPRO | Nosso software |
| Apresenta Cronograma da produção | Sim | Não | Sim |
| Traking de encomenda | Sim | Sim | Sim |
| Calcular custos de produção | Sim | Não | Sim |
| Calcular tempo de entrega | Sim | Não | Sim |
| Ver estado da produção em tempo real | Sim | Sim | Sim |
| Visualizar responsável por cada etapa da produção | Não | Não | Não |
| Informar atrasos | Não | Não | Sim |
| Relatório detalhado de produção | Não | Sim | Não |
| Pedido de materiais necessários ao stock | Não | Sim | Sim |
| Histórico de materiais usados no processo de fabrico. | Sim | Sim | Sim |
| Apresenta todos os passos necessários para produzir a encomenda | Sim | Sim | Sim |

Tabela 2.2.3 Tabela de funcionalidades

# Diagrama de Casos de Uso

Depois de apresentado no tópico anterior os atores que fazem parte do sistema, bem como

os objetivos e o papel de cada um, segue-se o diagrama de casos de uso. Um diagrama de

casos de uso mostra os casos de uso, atores e as suas interações. Na Figura 3.1, encontra-se ilustrado num diagrama os casos de uso e os respetivos atores. Este serve para mostrar

todas as funcionalidades do tema que escolhemos e para observar o acesso a todas as funcionalidades.

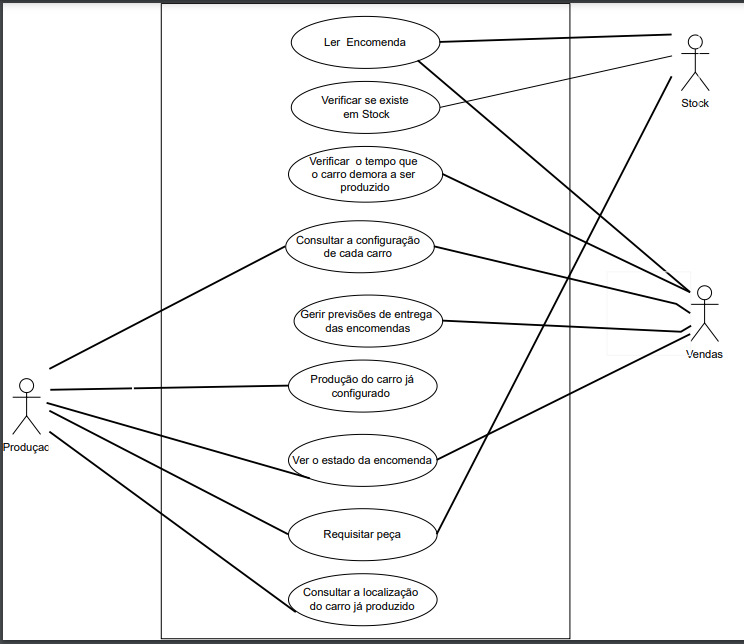


Figura 3.1 Casos de uso

3.1 Descrição dos casos de uso

O próximo passo a ser efetuado depois da identificação da forma como os atores irão interagir com a aplicação irá ser documentar os casos de uso. Deverá ser detalhado aquilo que o sistema tem de fornecer ao ator quando o caso de uso é executado, mostra também como o caso de uso começa e termina. Como aspeto de chave para a compreensão deste conceito, convém ter em atenção que um caso de uso não é um módulo de software – é antes algo que fornece valor ao ator. A descrição de um caso de uso mostra a razão da necessidade do Sistema. São muito úteis para auxiliar na análise de requisitos do sistema. Cada caso de uso corresponde a um requisito potencial, sendo por isso essencial para se obter uma visão mais aprofundada de como o utilizador fará a interação com a aplicação.

O que tem de ter a descrição de um caso de uso?

Por um lado, o fluxo normal de eventos (happy path), em que tudo corre bem. Por outro lado, existe o fluxo de eventos anormal, em que as coisas não correm de forma normal. Cada descrição de caso de uso é constituída pelos seguintes tópicos (template):

• **Nome**: nome do caso de uso que se irá descrever;

• **Descrição**: descrição curta e sucinta do caso de uso em questão. Deverá ser

percetível o que se pretende numa curta frase;

• **Pré-Condição**: condição inicial necessária para que o caso de uso decorra com

sucesso;

• **Caminho Principal**: descrição de como o utilizador deve proceder para que tudo

corra com sucesso;

**• Caminhos Alternativos**: descrição do que poderá correr mal em determinado

passo do caminho principal;

• **Pós-Condição**: condição em que se encontra o sistema após o término deste caso

de uso;

• **Suplementos ou adornos:** descrição de testes a realizar, requisitos não

funcionais.

As secções seguintes descrevem os principais dos casos de uso com o template apresentado

# 3.2 Gerir previsões da entrega das encomendas

Quando o ator (Vendas) interage com o planeamento da produção com base nas encomendas desencadear uma sequência de eventos que permitem desenvolver este caso de uso. A Tabela 4.2 descreve o processo que o autor executa para conseguir gerir as encomendas.

|  |  |
| --- | --- |
| **Nome** | Gerir previsões da entrega das encomendas |
| **Descrição** | Este caso de uso tem como objetivo descrever o processo para Gerir previsões da entrega das encomendas |
| **Pré-condição** | *Assegurar que o administrador tem privilégios para aceder as encomendas* |
| **Caminho Principal** | 1. O administrador consegue altera a data das encomendas 2. O sistema guarda a alteração 3. O sistema informa o cliente que a data da encomenda foi alterada |
| **Caminho Alternativos** | 1.a) O administrador não tem privilégios suficientes para podes gerir as encomendas  4.a) O sistema não consegue avisar o cliente por falta de  credenciais |
| **Pós-Condição** | O administrador precisa de ter privilégios suficientes para poder gerir as encomendas |
| **Suplementos ou adornos** | Testar se reunimos todas as informações necessárias para pode avisar o cliente em caso de alteração da data da sua encomenda |

# 

Tabela 4.2 Descrição de Casos de Uso" Gerir previsões da entrega das encomendas"

# 3.3 Diagrama de Sequencia

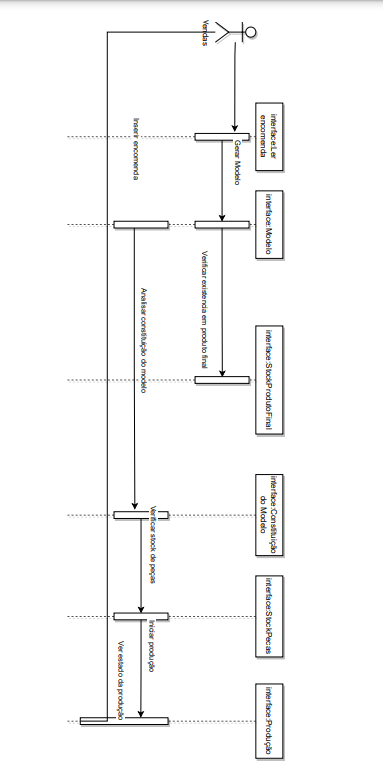


Figura 2-Diagrama de Sequencias

Um diagrama de sequência consiste em um grupo de objetos representados por linhas de vida e as mensagens que eles trocam durante a interação. Um diagrama de sequência mostra a sequência de mensagens transmitidas entre objetos

# User Stories

Uma User Storie é uma explicação informal e geral de uma funcionalidade na perspetiva do utilizador final ou cliente. Como ilustrado na Figura abaixo, podem-se ver as User Stories que decidimos aplicar no nosso trabalho.

|  |  |  |
| --- | --- | --- |
| **Epic 1** | **User Story** | **Acceptance Criteria** |
| **Como gestor eu quero ter a possibilidade de atribuir tarefas a um colaborador** | Assegurar que o admin tem privilégios para aceder as encomendas | * Acesso a lista de encomendas * Consulta estado da encomenda * Consulta a quantidade das peças de cada encomenda * Consultar o estado encomendas |
| Assegurar que o admin tem privilégios para consultar a produção | * Acesso a lista de produção * Consultar a lista de produção * Consulta estado da produção |
| Como admin quero ver todas as encomendas | * Acesso a lista de encomendas * Consultar a lista de encomendas * Consulta estado da encomenda * Consulta estado da produção |

|  |  |  |
| --- | --- | --- |
| Epic 1 | User Story | Acceptance Criteria |
| Como Engenheiro mecânico eu quero ter a possibilidade de associar peças a carros, no entanto, não quero que outros funcionários o façam, no entanto eles podem ver as associações entre as peças e os carros | Assegurar que só o engenheiro mecânico tem privilégios suficientes para criar associações entre peças e carros. | * Acesso a página de Criar das associações entre Carros e peças. |
| Assegurar que só o engenheiro mecânico tem privilégios suficientes para editar associações entre peças e carros. | * Acesso a página de Edit das associações entre Carros e peças. |
| Assegurar que só o engenheiro mecânico tem privilégios suficientes para eliminar associações entre peças e carros. | * Acesso a página de Delete das associações entre Carros e peças. |
| Assegurar que para além do engenheiro mecânico, os restantes colaboradores têm privilégios suficientes para ver as associações entre peças e carros. | * Acesso a lista de associações entre carros e peças. |

Tabela 4.1 Tabela de User Stories

# Diagrama de Classes

Uma classe é uma descrição de um conjunto de objetos (um objeto é uma instância de uma classe, isto é, uma manifestação concreta de algo abstrato) que partilham os mesmos atributos, operações, relacionamentos e semântica.

Nesta fase do projeto foi desenvolvido o diagrama de classes apresentado na Figura.

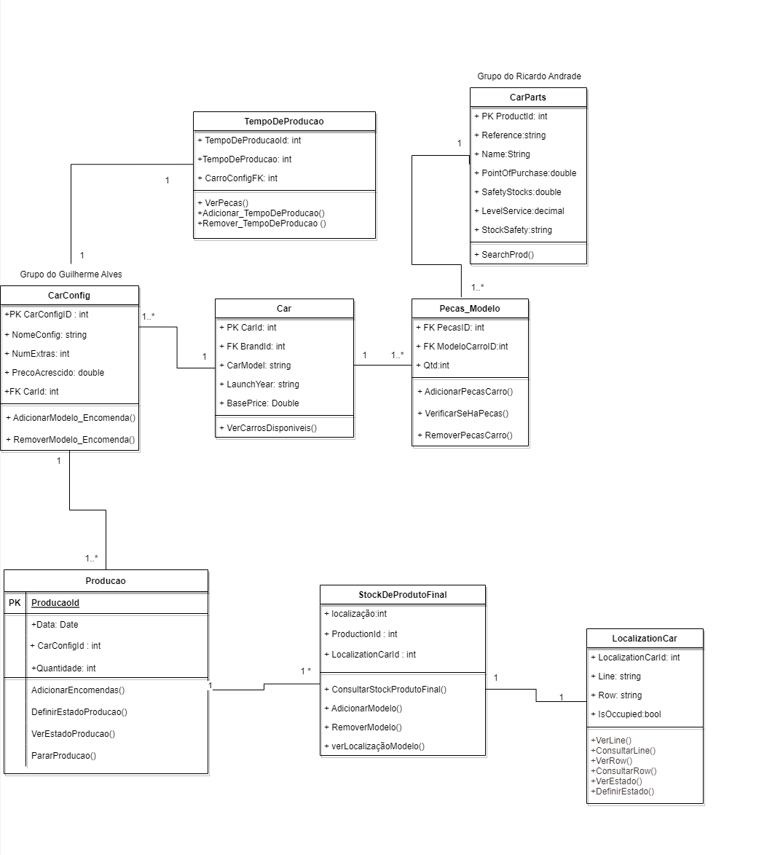


Figura 5.1 Diagrama de classes

Uma imagem com mesa

Descrição gerada automaticamente

Tabela 5.1 Classe Pecas

Uma imagem com mesa

Descrição gerada automaticamente

Tabela 5.2 Tabela Modelo\_Carro

Uma imagem com mesa

Descrição gerada automaticamente

Tabela 5.3 Tabela Modelo\_Carro\_Pecas

Uma imagem com mesa

Descrição gerada automaticamente

Tabela 5.4 Tabela Modelo\_Carro\_Encomenda

Uma imagem com mesa

Descrição gerada automaticamente

Tabela 5.5 Tabela Operacao inserir na tabela Pecas

Uma imagem com mesa

Descrição gerada automaticamente

Tabela 5.6 Tabela Operacao inserir na tabela Modelo\_Carro\_Pecas

Uma imagem com mesa

Descrição gerada automaticamente

Tabela 5.7 Operacao inserir na tabela Modelo\_Carro

# Diagrama de Estados

Um diagrama de estados mostra os eventos que causam transição de um estado para o outro, assim como as ações que resultam de uma alteração de estado. Um evento é uma ocorrência significativa que tem uma localização no tempo e no espaço.

Na Figura, mostra-se o diagrama de estados da encomenda.

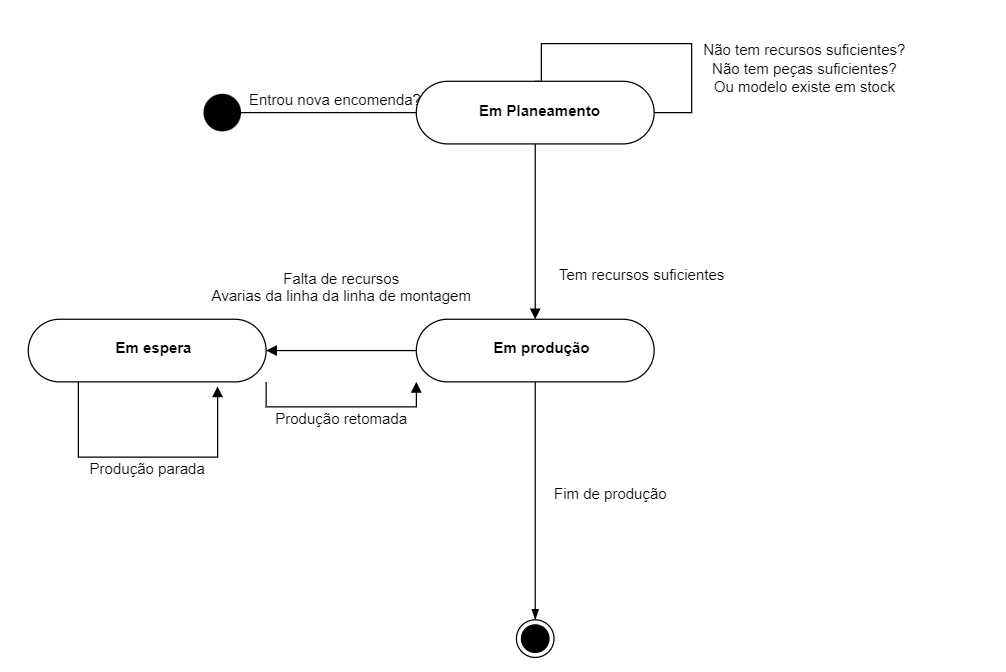


Tabela 6.1 Diagrama de estados

Uma imagem com mesa

Descrição gerada automaticamente

Tabela 6.2 Classe produção

# Diagrama de Pacotes

O Diagrama de pacotes, ou diagrama de módulos, definido pela UML, descreve os pacotes ou pedaços do sistema divididos em agrupamentos lógicos mostrando as dependências entre eles. Este diagrama é muito utilizado para ilustrar a arquitetura de um sistema mostrando o agrupamento de suas classes. Um pacote representa um grupo de classes (ou outros elementos).

Nesta fase foi realizado o diagrama de pacotes apresentado na Figura.

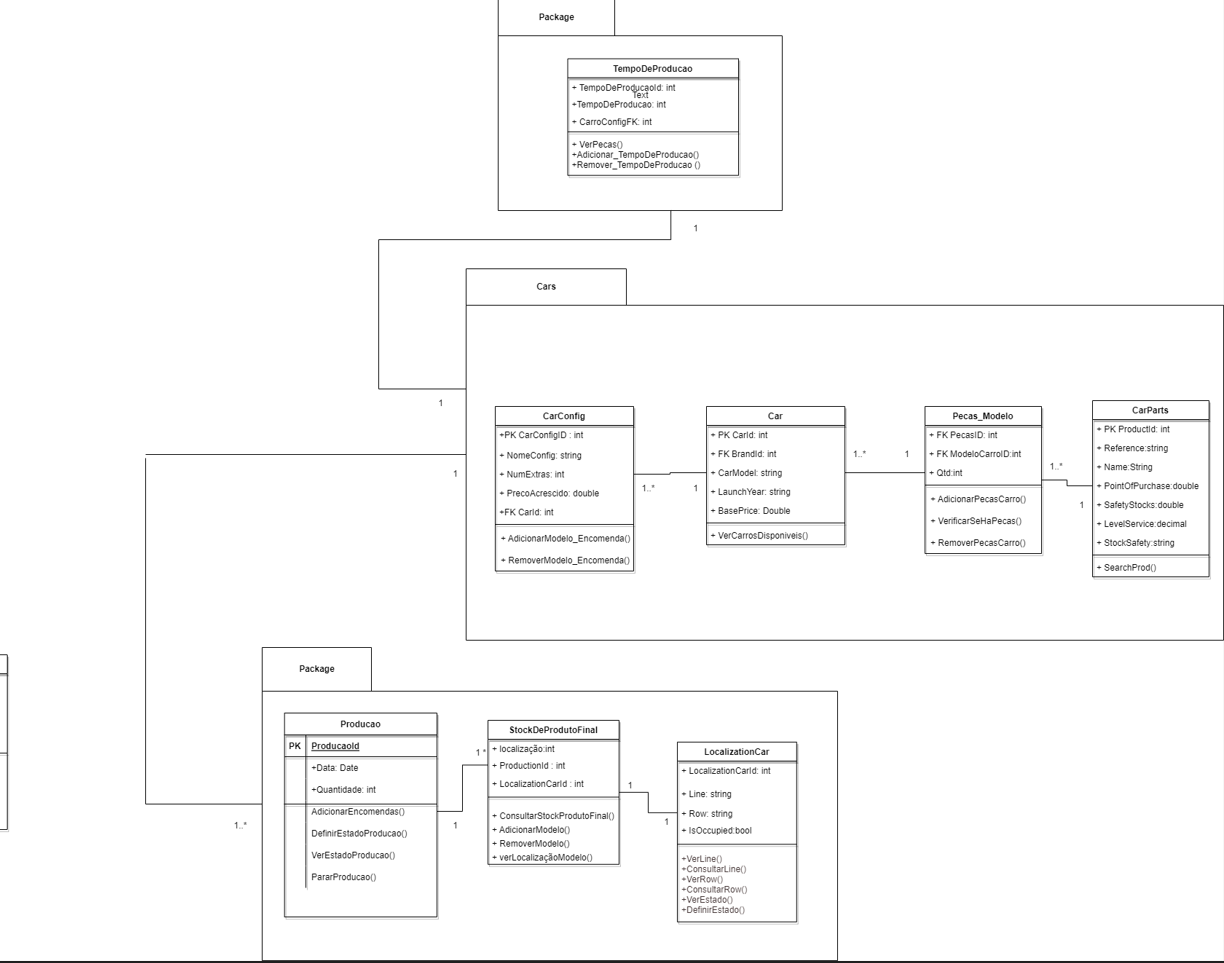


Figura 7.1 Diagrama de Pacotes

Diagrama de Componentes

Os diagramas de componentes são utilizados para ilustrar as dependências entre componentes de software, incluindo componentes de código fonte ou executáveis. Apresentam os componentes que compõem uma aplicação, sistema ou empresa. São apresentados os componentes, as suas inter-relações, interações e interfaces públicas. Os componentes encontram-se interligados por uma relação de dependência para mostrar o impacto nos diversos componentes das alterações de um componente em particular.

Nesta fase do projeto foi desenvolvido o diagrama de componentes apresentado na Figura abaixo.

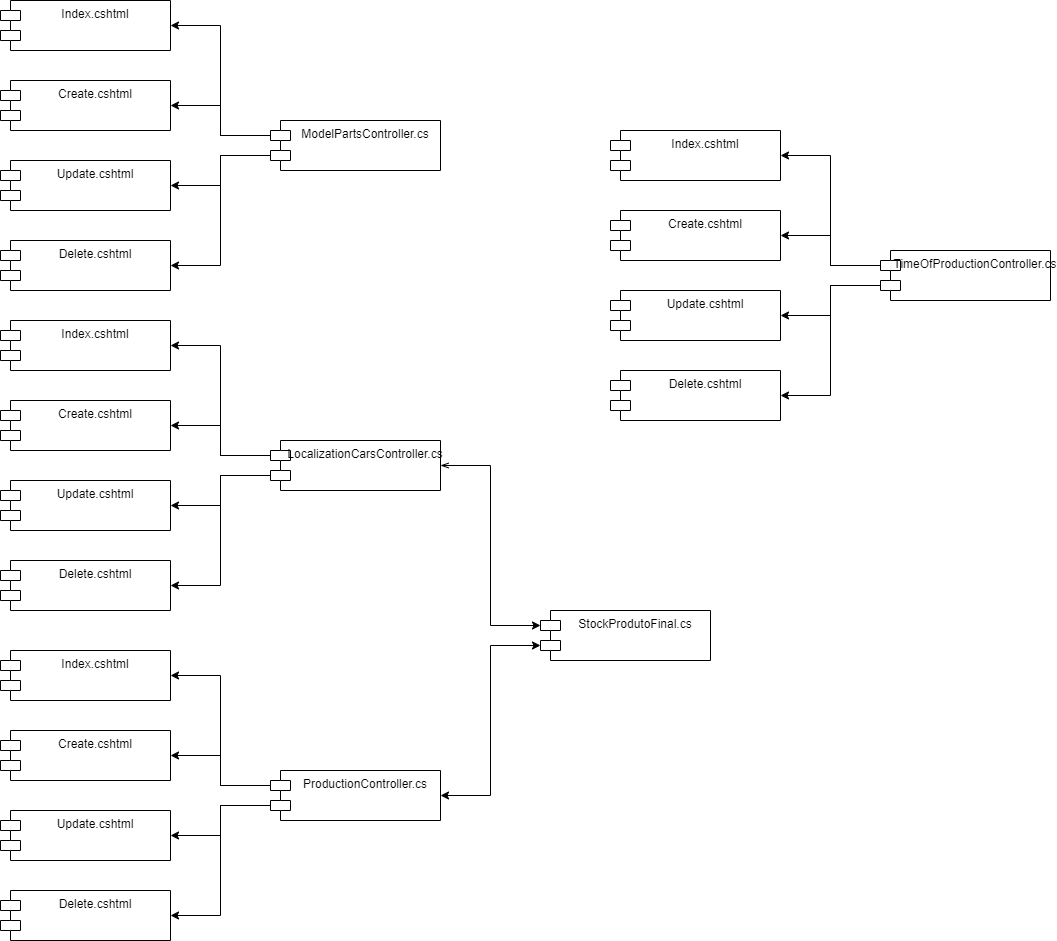


Figura - Diagrama de Componentes

# Diagramas de Instalação

Os diagramas de instalação ilustram a arquitetura do sistema em termos de nós (nodes) que efetuam o processamento de componentes. Permite mostrar como o hardware estará organizado e como os componentes (software) estarão distribuídos.

Nesta fase do projeto foi desenvolvido o diagrama de instalação apresentado na Figura x.

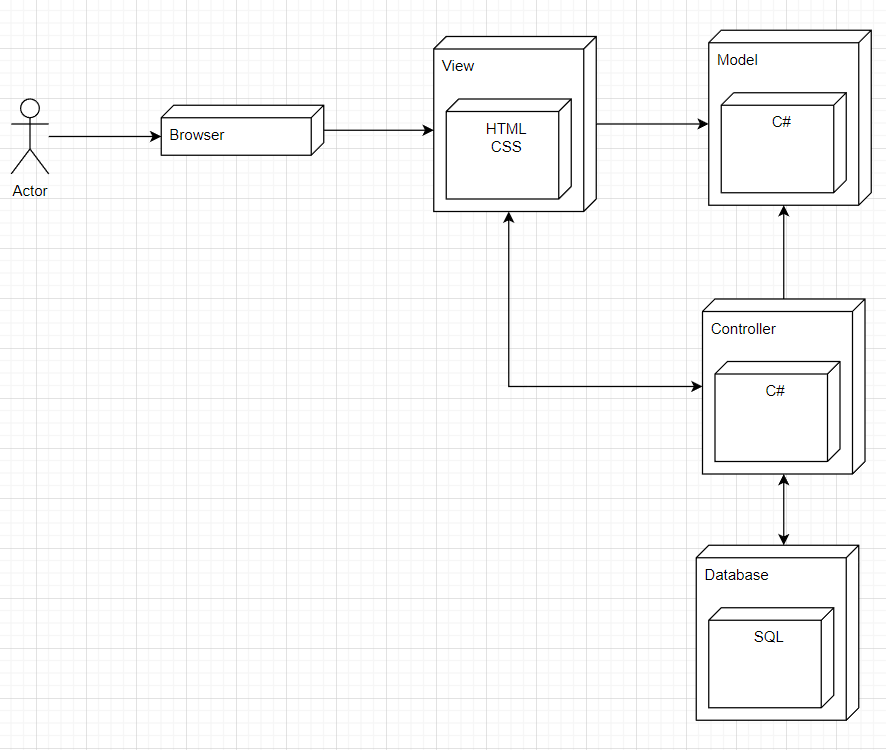


Figura 8.1 Diagrama de instalação

# Tabelas de Casos de Teste

|  |  |  |  |
| --- | --- | --- | --- |
| **ID do teste** | **O que vamos testar** | **Valor introduzido** | **Valor esperado** |
| **ID1** | Introduzir peça | n/d | Por favor, introduza uma peça |
| **ID2** | Introduzir model | n/d | Por favor, introduza um modelo |
| **ID3** | introduzir peça em modelo sendo que ambos já se encontram na tabela | modelo, peça | O modelo e peça introduzida já se encontram relacionados |

Tabela 9.1 Tabela de casos de teste inserir novo modelo e peça na tabela ModelParts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inserir um modelo de Carros | | | | |
| **ID do Teste** | **o que vamos testar** | **Nome** | **DeadLine** | **Resultado Esperado** |
| ID1 | Verificar se o modelo já foi criado | V | X | " THE MODEL IS REQUIED" |
| ID2 | Verificar se a quantidade do modelo | V | X | " The values is valid" |
| ID3 | DeadLine ( o modelo não foi selecionado ) | X | V | "DeadLine the model not selected" |
| ID4 | Todos os valores estão corretos | X | X | A Tarefa é criada |

Tabela 9.2 Tabela de casos de teste inserir novo modelo de Carros na tabela CarModels

# Protótipos

Nesta parte do projeto será possível ver alguns dos ecrãs presentes na nossa aplicação.

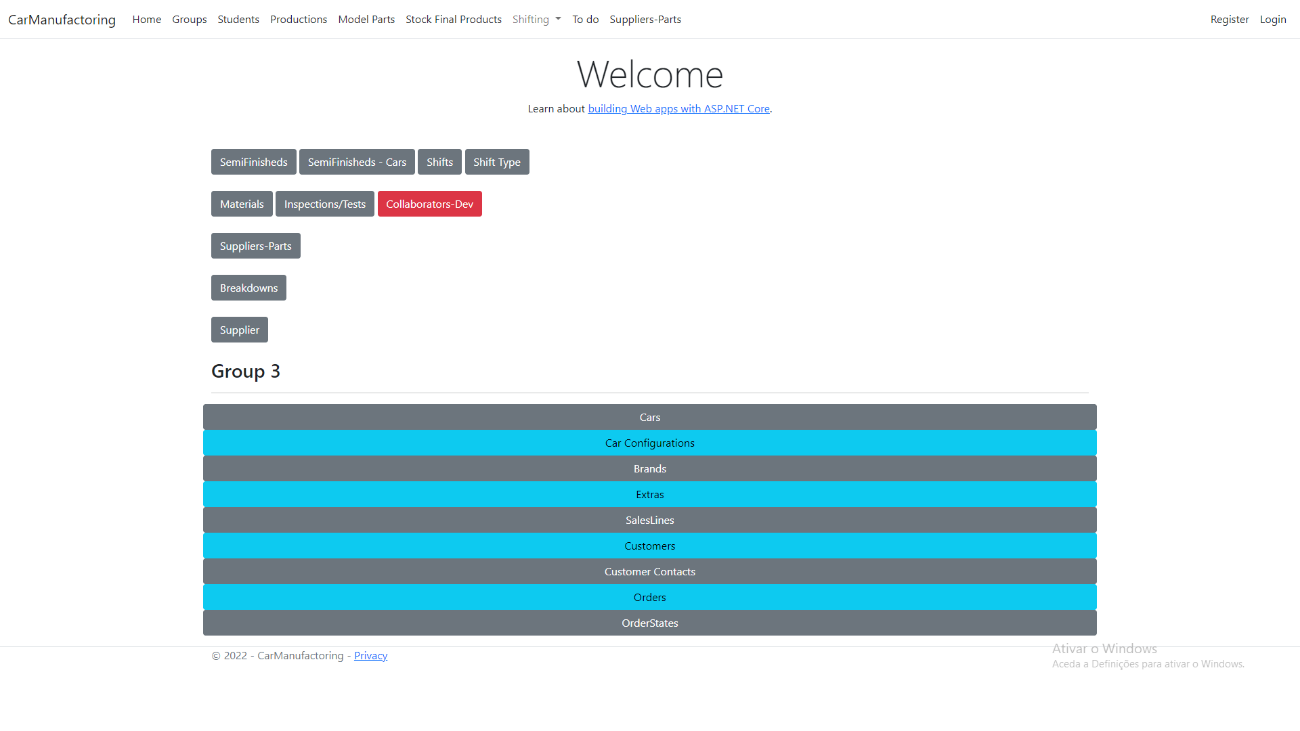


Figura -Página Inicial do projeto

Uma imagem com texto

Descrição gerada automaticamente

Figura -Pagina Login do Projeto

Uma imagem com mesa

Descrição gerada automaticamente

Figura -Pagina StockFinalProduct

Uma imagem com mesa

Descrição gerada automaticamente

Figura -Pagina ModelParts

# Conclusão

Ao realizar este trabalho conseguimos ter uma melhor visão sobre o quão difícil é o desenvolvimento de um Software ERP e a juntou-nos também a perceber o quão difícil é desenvolver um software de grande dimensão.

O maior obstáculo que enfrentamos durante todo o processo foi a falta de comunicação e organização entre grupos, para alem disso durante grande parte do tempo houve diversas incoerências entre os diagramas de classes de todos os grupos.

Concluindo, acreditamos ter tido sucesso na elaboração do trabalho sendo que todos os pontos definidos pelos docentes, foram realizados.

# Referencias Biográficas

Silveira, M. C . Apontamentos Engenharia Software II.

Ovelheiro. B.S.R Relatório para a obtenção do grau de licenciado em engenharia informática

Guia Gastronómico -Relatório Engenharia Software 2020

# Anexos

Trelo Inicial

Uma imagem com texto

Descrição gerada automaticamente

Figura 1.13.1 Trello do grupo https://trello.com/b/BjVoUXpE/planejar-produ%C3%A7%C3%A3o-com-base-nas-encomendas

Uma imagem com texto

Descrição gerada automaticamente

Figura 13.2 Trello do grupo (Cont.) <https://trello.com/b/BjVoUXpE/planejar-produ%C3%A7%C3%A3o-com-base-nas-encomendas>

Trello Final



Figura - Trello Final

GIt Hub

Uma imagem com texto

Descrição gerada automaticamente

Figura -Pagina do GitHub - https://github.com/noellopes/CarManufactoring

# Autoavaliação

Engenharia de Software II

Rui Condesso 1701429 – 13 Valores

Ricardo Sousa 1705109 – 13 Valores

Programação para a Internet

Rui Condesso 1701429 – 11 Valores

Ricardo Sousa 1705109 – 13 Valores

Pedro Matos 1700789 – 12 Valores