



Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2024/2025

Fábrica de cadeiras

Marco Soares Gonçalves (a104614) André Filipe Soares Pereira (a104275)
Salvador Duarte Magalhães Barreto(a104520) Leonardo Gomes Alves(a104093)

26 de Janeiro de 2025

L | 4

Data de Receção	
Responsável	
Avaliação	
Observações	

Fábrica de cadeiras

Marco Soares Gonçalves (a104614) André Filipe Soares Pereira (a104275)
Salvador Duarte Magalhães Barreto(a104520) Leonardo Gomes Alves(a104093)

26 de Janeiro de 2025

Resumo

O presente relatório tem como objetivo demonstrar o nosso trabalho desenvolvido para o projeto de Laboratórios de Informática IV.

O nosso trabalho aborda detalhadamente o desenvolvimento de um sistema de gestão de uma fábrica de cadeiras chamada "Cadeiras Lusitanas", fundada por Inácio, com o objetivo de criar cadeiras ergonómicas para melhorar a qualidade de vida daqueles que ficam grande parte dos seus dias sentados.

Este sistema deverá suportar funções como: suportar a utilização por diversos utilizadores; registar a quantidade de stock de um certo produto ou componente; dar entrada e saída de stock; ser possível verificar o tempo restante para a conclusão da produção dos produtos que estão a ser produzidos no momento.

Durante a conceção do sistema, foram realizadas diversas reuniões com o nosso cliente com o objetivo de recolher requisitos fundamentais à construção de um bom sistema. Com essas informações, foram organizados requisitos que, após validados com o cliente, ajudaram a construir um modelo de domínio, diagramas de Use Case e as suas especificações e um diagrama de atividades. O modelo e os diagramas foram fundamentais para haver um alinhamento das ideias do cliente e da equipa de desenvolvimento.

Foram ainda apresentados diversos esboços das interfaces do sistema, respeitando sempre as preferências do cliente, que solicitou um design simples, funcional e intuitivo. Estes esboços incluem, por exemplo, telas para iniciar sessão, gestão de produtos e componentes e visualização da linha de montagem dos produtos.

Na seguinte fase, a aplicação foi desenvolvida utilizando tecnologias como C#, .NET, Blazor, Microsoft SQL Server e Docker, seguindo sempre os requisitos pré-estabelecidos na primeira fase e realizando reuniões frequentes entre o cliente e a equipa de desenvolvimento.

Desta forma, obtemos uma aplicação bastante parecida com a prototipada nos *mock-ups*, com a devida implementação de todas as funções idealizadas e apenas as alterações necessárias para melhorar a usabilidade.

Área de Aplicação: Desenvolvimento de Sistemas de Gestão; Engenharia de Software.

Palavras-Chave: desenvolvimento de "software"; linha de montagem; C#; .NET; bases de dados relacionais; UML; Blazor; Docker.

Índice

1	Introdução	1
1.1	Contextualização	1
1.2	Motivação e objetivos	1
1.3	Justificação e utilidade do sistema	2
1.4	Estabelecimento da identidade do projeto	2
1.5	Identificação dos recursos necessários	3
1.6	Maqueta do sistema	4
1.7	Definição de um conjunto de medidas de sucesso	5
1.8	Plano de desenvolvimento (diagrama GANTT)	5
2	Levantamento e Análise de Requisitos	7
2.1	Apresentação da estratégia e método	7
2.2	Requisitos funcionais	8
2.3	Requisitos não funcionais	17
2.4	Validação dos requisitos estabelecidos	17
3	Especificação e modelação do "software"	18
3.1	Apresentação geral da especificação	18
3.2	Aspetos estruturais	19
3.3	Aspetos comportamentais	22
3.3.1	Diagramas de Use Case	22
3.3.2	Use Case	26
3.3.3	Diagrama de atividades	34
4	Conceção do Sistema de Dados	36
4.1	Apresentação geral da estrutura do sistema de dados	36
4.2	Descrição detalhada dos vários elementos de dados e seus relacionamentos	37
5	Esboço das Interfaces do Sistema	40
5.1	Estrutura geral das interfaces do sistema	40
5.2	Caracterização das interfaces	45
6	Implementação da Aplicação	47
6.1	Apresentação e descrição do processo de implementação realizado.	47
6.2	Apresentação da aplicação e explicação dos serviços implementados.	52
6.3	Análise e avaliação da aplicação desenvolvida.	56

7 Conclusão	58
8 Referências	59
9 Sinónimos	60

Lista de Figuras

1.1	Maqueta do sistema	4
1.2	Diagrama GANTT proposto inicialmente	6
1.3	Diagrama GANTT real	6
3.1	Modelo de Domínio	20
3.2	Diagrama de Use Case do Sistema	23
3.3	Diagrama de Use Case do subsistema de componentes.	24
3.4	Diagrama de Use Case do subsistema de produtos	25
3.5	Diagrama de atividades	35
4.1	Modelo Lógico	36
5.1	Menu "Iniciar sessão"	40
5.2	Menu "Página inicial"	41
5.3	Menu "Catálogo de Produtos"	41
5.4	Menu "Catálogo de Componentes"	42
5.5	Menu "Criar novo Componente"	42
5.6	Menu "Criar novo Produto"	43
5.7	Menu "Visualizar linha de Montagem"	43
5.8	Menu "Criar novo Utilizador"	44
5.9	Menu "Inserir Identificador"	44
5.10	Menu "Inserir Quantidade"	44
6.1	Arquitetura Inicial	47
6.2	Pastas do projeto	48
6.3	Funcionamento da linha de montagem	50
6.4	Arquitetura final	51
6.5	Página Inicial	52
6.6	Catálogo de Produtos	53
6.7	Linha de Montagem	54
6.8	Catálogo de Componentes	55

Lista de Tabelas

1.1	Recursos Humanos	3
1.2	Recursos Materiais	3
4.1	Entidade Componente	37
4.2	Entidade Produto	37
4.3	Entidade Instância de Produto	38
4.4	Entidade Produto_tem_Componente	38
4.5	Relacionamento Produto-Instância de Produto	39
4.6	Relacionamento Utilizador-Instância de Produto	39
4.7	Relacionamento Produto_tem_Componente	39

1 Introdução

1.1 Contextualização

Um agora ex-funcionário de uma grande empresa de desenvolvimento de "software" ("Lusitana Tech"), começou a sentir regularmente dores intensas nas suas costas. Face a este problema decidiu falar com a sua médica de família para descobrir a origem. No consultório, a médica sugeriu que este provavelmente se devesse ao facto de o seu trabalho requerer horas a fio sentado numa cadeira que não oferece suporte ergonómico adequado. Para chegar ao fundo deste problema, decidiu então criar um inquérito na dita empresa, inquérito este que perguntava aos seus funcionários se também experienciaram as dores de costas frequentes. Para sua surpresa, a grande maioria dos votos confirmou o pior. Inácio sempre quis criar a sua própria empresa e agora, com muita experiência de trabalho e um interesse recém-adquirido pelos benefícios ergonómicos, decidiu estabelecer a sua própria produtora de cadeiras, "Cadeiras Lusitanas".

1.2 Motivação e objetivos

O Inácio sabia que se queria levar este projeto a sério necessitaria de conseguir gerir toda a produção e o "stock" do seu material. No entanto, para além de ele não possuir os fundos para contratar um secretário que fizesse este trabalho por si, ele sabe que com o eventual crescimento da empresa impossibilitar-se-ia gerir tudo manualmente. Decide então aplicar a sua experiência na área do desenvolvimento de "software" para criar um programa que lhe garanta uma solução barata e escalável para este problema, visando:

- Diminuir custos administrativos e de produção;
- Maximizar velocidade e capacidade de produção;
- Minimizar erros humanos;
- Adquirir estatísticas relevantes para otimizar as linhas de montagem.

1.3 Justificação e utilidade do sistema

Apesar de a utilização de um "software" obrigar a compra e instalação de aparelhos particulares e dispendiosos, o retorno sobre o investimento será garantido pela eficiência de produção e redução de desperdício material a longo prazo; além disso, facilitamos a futura expansão das linhas de produção com a fácil escalabilidade de um "software" moderno. Os dados estatísticos coletados podem por ventura permitir a implementação de melhorias nos processos de montagem. A aplicação irá garantir diversas utilidades pertinentes como:

- Iniciar facilmente a montagem de cadeiras;
- Acompanhar em tempo real o estado da produção de cadeiras;
- Verificar facilmente as quantidades de "stock" tanto de cadeiras (produto) como de componentes;
- Recolher estatísticas relevantes acerca da linha de montagem.

1.4 Estabelecimento da identidade do projeto

O projeto "**Cadeiras Lusitanas**" surge com um propósito ambicioso. Mudar a vida de milhões de pessoas que devido aos seus ambientes de trabalho necessitam de passar horas a fio sentadas em cadeiras inadequadas para as suas costas. A importância de uma boa cadeira ergonómica tem sido cada vez mais discutida e com este projeto, pretende-se revolucionar como estas cadeiras são construídas, focando sempre a sua criação em estudos científicos e naquilo que melhor satisfaz os consumidores.

Nome do projeto: Cadeiras Lusitanas

Cliente: Inácio(Futuro CEO da Cadeiras Lusitanas)

Tempo de execução esperado: 3 meses

Data de início: 16/09/2024

Visão e missão: Este projeto tem como missão garantir que pessoas que necessitem de passar muitas horas em sentadas num escritório, o possam fazer sem grandes consequências para a saúde das suas costas. Inicialmente estas cadeiras estarão apenas disponíveis para a "Lusitana Tech", mas é esperado que se expanda o negócio a uma escala mundial.

Riscos:

- **"Feedback" do cliente:** As cadeiras podem não corresponder às expectativas do cliente sobre conforto e ergonomia.
- **Custos inesperados:** A qualquer momento os componentes da cadeira podem subir de preço, tornando a continuação do projeto inviável.
- **Problemas legais:** Ainda que tenha sido realizada muita consulta sobre normas a ser seguidas na criação de cadeiras, podem existir normas diferentes em certas regiões do mundo, o que obrigaria a reuniões sobre a importância de criar produtos que possam ser vendidos nesses locais.

Critério de sucesso: O projeto será um sucesso se nos próximos 15 anos se verificar uma redução nos problemas de saúde dos utilizadores do produto e se as vendas do produto superarem as cinquenta mil vendas.

1.5 Identificação dos recursos necessários

Para desenvolver a aplicação é necessária uma colaboração contínua entre Inácio e uma equipa de desenvolvimento qualificada. Além disso, foi determinado o "software" a utilizar e o "hardware" requisitado para o executar. Seguem-se as especificações dos recursos humanos e materiais requisitados.

Recursos Humanos
Inácio (futuro CEO da empresa e dirigente do projeto)
Equipa de desenvolvimento de "software" composta por 4 elementos

Tabela 1.1: Recursos Humanos

Recursos Materiais	
"Software"	Hardware
Visual Studio, .NET C#, SQL Server e Office Blazor Github MySQL Workbench, Docker Krita, Overleaf Copilot	5 computadores pessoais (1 para cada um dos intervenientes humanos a trabalhar no projeto) 1 servidor (para preservar os dados na fábrica da "Cadeiras Lusitanas")

Tabela 1.2: Recursos Materiais

1.6 Maqueta do sistema

Durante a primeira reunião com Inácio ficamos a perceber que este não procura por algo muito ambicioso, quer apenas um programa que lhe permita gerir os seus produtos e componentes, que proporcione uma interface simples e agradável e que requeira pouco conhecimento sobre tecnologia.

Tendo tudo isto em conta, decidimos criar a seguinte maqueta de modo a fornecer uma interpretação inicial daquilo que o "software" pode representar:

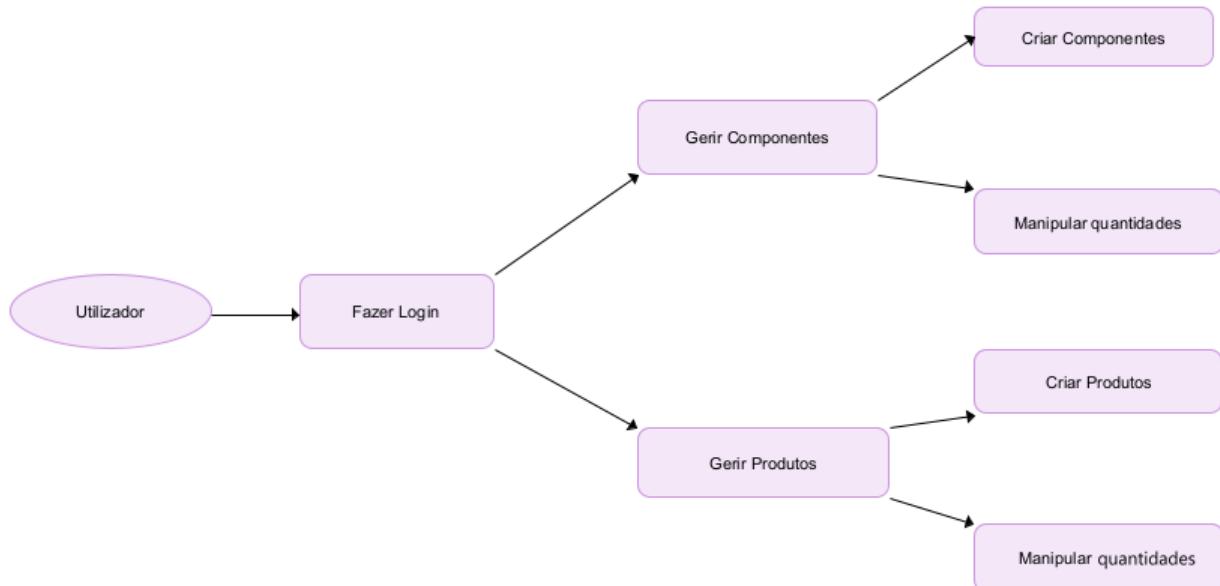


Figura 1.1: Maqueta do sistema

Atendendo à maqueta apresentada acima, um utilizador terá de fazer "login", iniciar sessão, para acessar a aplicação, e após isto, ele poderá escolher entre gerir produtos ou componentes. Dependendo da sua escolha terá então acesso a diferentes funcionalidades de manipulação e criação.

1.7 Definição de um conjunto de medidas de sucesso

Para garantir a utilidade e valor da aplicação para os funcionários que virão a utilizá-la determinamos as principais metas a cumprir:

- **Satisfação de utilizadores:** através do "feedback" do dirigente do projeto com a colaboração de futuros funcionários da fábrica da "Cadeiras Lusitanas";
- **Cumprimento de prazos:** foram definidos prazos a cumprir que serão um indicador importante da eficiência e cuidado da equipa de desenvolvimento na produção do "software";
- **Desempenho da Aplicação:** será imperativo garantir um nível de desempenho respeitável que garanta uma operação agradável aos utilizadores com tempos de resposta mínimos.
- **Cumprimento dos Requisitos:** como principal meta é absolutamente necessário garantir que todos os requisitos funcionais e não funcionais definidos sejam cumpridos sempre que possível.

1.8 Plano de desenvolvimento (diagrama GANTT)

Para garantir o sucesso da "Cadeiras Lusitanas" foi criado um diagrama "GANTT" de modo a representar os prazos de cada etapa da modelação do projeto. Desta forma tentamos definir claramente todas as etapas do trabalho, permitindo um fácil acompanhamento do desenvolvimento, bem como a definição de metas claras e alcançáveis.

A fase inicial do projeto ficou dividida em **6 etapas**:

- **Introdução;**
- **Levantamento e Análise de Requisitos;**
- **Especificação e Modelação do "Software";**
- **Conceção do Sistema de Dados;**
- **Esboço das Interfaces do Sistema;**
- **Conclusões e Trabalho Futuro.**

Para a 1.^a etapa, tendo em conta a sua simplicidade, decidimos que não seria necessário investir uma quantidade excessiva de tempo, pelo qual se estabeleceu um prazo de 6 dias.

Tomamos a decisão de investir a maior carga horária do projeto na 2.^a e 3.^a etapas, pois a nosso ver, estas etapas são a base de todo o nosso trabalho e necessitam de ser cuidadosamente pensadas para evitar imprevistos na fase seguinte. Os prazos definidos foram de 20 e 22 dias, respectivamente.

Na 4.^a etapa, dada a experiência prévia da equipa com Bases de Dados, foi estabelecido um prazo de apenas 10 dias, já que se espera uma grande eficácia no seu desenvolvimento.

A 5.^a etapa reserva 10 dias de modo a possibilitar "mockups" que sejam precisos e representem menus o mais próximo possível aos que serão implementados.

Finalmente, para a última etapa foi reservado apenas 1 dia, pois todo o trabalho já deverá estar realizado e apenas será necessário deixar algumas reflexões sobre como esta fase correu e sobre o que deverá ser realizado na próxima.

FÁBRICA DE CADEIRAS

LEI - Laboratórios de Informática IV - 2024/2025
Grupo 15

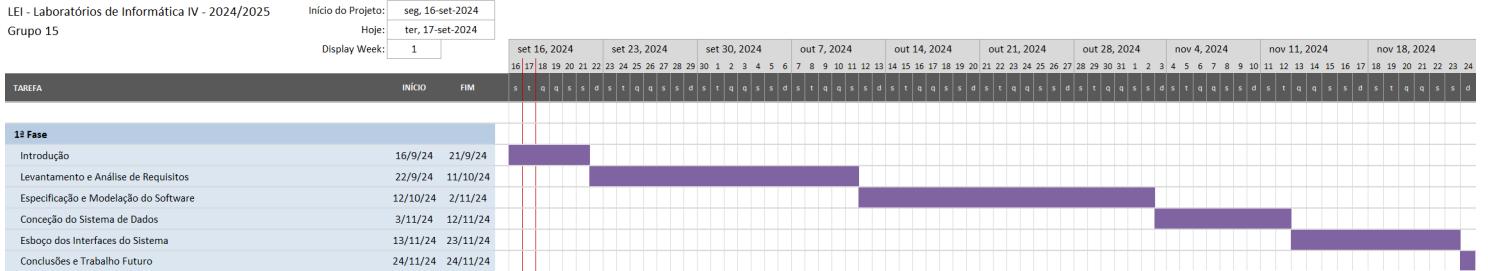


Figura 1.2: Diagrama GANTT proposto inicialmente

Apesar de, numa fase inicial, a divisão do tempo entre as etapas parecer adequada, à medida que a equipa foi realizando o trabalho, percebemos, logo na 2.^a fase, que o tempo atribuído não correspondia com a realidade, nomeadamente pelo facto de estarmos sempre em constante revisão e correção. Por esse motivo, a 2.^a e 3.^a etapas foram prolongadas alguns dias, o que nos levou a reduzir o tempo da 4.^a etapa pois, como referido anteriormente, já possuímos uma maior experiência no trabalho que seria feito. Apesar de todos os atrasos, conseguimos atingir os nossos objetivos.

FÁBRICA DE CADEIRAS

LEI - Laboratórios de Informática IV - 2024/2025
Grupo 15

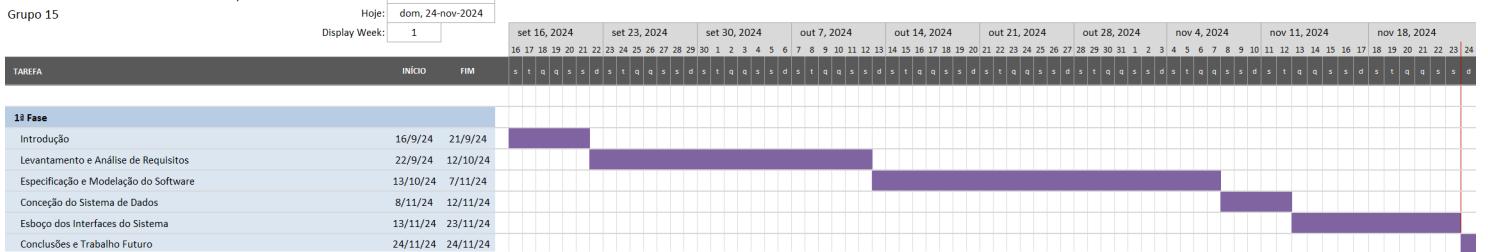


Figura 1.3: Diagrama GANTT real

2 Levantamento e Análise de Requisitos

2.1 Apresentação da estratégia e método

No contexto de levantamento de requisitos encontramo-nos numa **posição** um tanto **privilegiada**, uma vez que muitos dos desafios que seriam comuns num projeto deste género não se aplicam ao nosso.

Primeiramente o nosso cliente apresenta um extenso **conhecimento** na área de desenvolvimento de "software", assim a **comunicação** entre todos nós pôde ser **fluida**, permitindo dialogar de forma a que todos os interlocutores envolvidos se **compreendessem equitativamente**, ou próximo disso. Pelo meio de jargão e conceitos relacionados ao processo de desenvolvimento conhecidos por todos, as **opiniões** de cada um foram **expostas e conciliadas** efetivamente.

Além disso, **Inácio** foi **específico, articulado e realista** na sua escolha de requisitos, por consequência conseguiu **minimizar** a **fricção** no processo de seleção dos requisitos relevantes, não só por isso, mas também, pelo fato de **Inácio** ser o **singular detentor** do **veredito** final no processo de escolha.

Dadas estas vantagens, listamos os **métodos concretos** de levantamento dos requisitos a que recorremos. Tanto nós como Inácio carecemos conhecimento essencial para a realização do projeto, portanto, para realizar um descobrimento de requisitos apropriado, visámos encontrar **fontes de informação fidedignas**, culminando em:

1. Modos de funcionamento de "software" conceptualmente **similar** e testado em cenários reais;
2. **Análise** ao vivo do ambiente de trabalho de uma **fábrica ativa**;
3. **Entrevistas** a candidatos que concorreram para posições na futura fábrica da "Cadeiras Lusitanas" (ex-funcionários de fábricas de cadeiras);
4. **Entrevista** a um gerente sénior de uma empresa produtora de cadeiras ergonómicas, o senhor Jorge Pereira.

Inicialmente foi realizada uma **reunião aberta** onde optamos pelos métodos de levantamento mencionados; através destes foi-nos possível **aprender** sobre o domínio da aplicação, **serviços**

a implementar e características a considerar durante a conceção dos requisitos.

Obter uma **perspetiva do dia a dia** no ambiente de trabalho de um gestor e de vários ex-funcionários da área foi excelente para **garantir** que as **necessidades** dos futuros **utilizadores** do sistema fossem cumpridas.

Tendo descrito explicitamente os **métodos** de levantamento de requisitos **adotados** e as vantagens inerentes de que se tirou proveito, tornamos evidente como o cliente e a equipa de desenvolvimento conseguiram produzir uma lista de **requisitos comprehensivos** e **consistentes**.

2.2 Requisitos funcionais

Para descrever os nossos requisitos optamos por uma **linguagem natural** simples de fácil compreensão. Tentamos evitar a ambiguidade, com o objetivo de facilitar o trabalho para ambas as partes envolvidas no projeto.

Usufruímos de um método de especificação onde cada requisito tem "**Requisitos de utilizador**" e "**Requisitos de sistema**"; os "**Requisitos de utilizador**" descrevem os serviços que devem ser cumpridos por parte do sistema e as restrições sob as quais devem cumpri-los. O nível de precisão que adotamos na descrição desses serviços é relativamente alto.

Os "**Requisitos de sistema**" são mais detalhados no sentido em que dispõe de uma descrição minuciosa das funções, serviços e restrições operacionais do sistema, com estes requisitos descreveremos precisamente aquilo que deve ser implementado com o cuidado de garantir consistência e zelar pelas metas futuras do projeto.

Para além dos requisitos de sistema e utilizador, vários requisitos incluem uma "**Fonte**" e um "**Motivo**" nas suas especificações. A "**Fonte**" de um requisito tem como propósito identificar a pessoa a quem se atribui a sugestão do requisito; o "**Motivo**" permite registar os fundamentos que justificam a relevância do requisito associado. Quase todos os requisitos incluem estes detalhes e, quando não é o caso, deve-se à indiscutível necessidade desses requisitos.

Vale ressaltar que qualquer referência a "**gestor**" na especificação dos requisitos refere-se ao registo específico do **Inácio**. Para além disto, os **catálogos** de componentes e produtos representam os componentes e produtos que vão estar **disponíveis** no sistema mas **não incluem todos** eles; é **possível** existirem produtos e componentes **descontinuados** que permanecem no sistema o que permite preservar históricos dependentes dos mesmo.

Segue-se uma enumeração e especificação de cada requisito funcional.

1 Criar novo utilizador

Fonte: Inácio na primeira reunião aberta.

Motivo: Inácio quer ser **detentor exclusivo** da capacidade de criar utilizadores no sistema; pretende que seja possível **criar os utilizadores** no sistema na sua conta exclusiva.

Requisitos de utilizador: O sistema deve permitir que o gestor (detentor exclusivo da capacidade de criar utilizadores) **crie um registo** de utilizador na aplicação de forma clara e intuitiva. Ao criar o novo registo, o sistema deve garantir que a sua integridade não será afetada e que não deverão existir conflitos. Esta operação deve ser eficiente e enviar "feedback" rápido ao gestor sobre o sucesso ou falha da mesma.

Requisitos de sistema:

1. O sistema deve **solicitar** ao gestor que introduza os **dados necessários** para realizar o registo (email e palavra-passe).
2. Após a introdução destes dados, o sistema deve **verificar** se o **email introduzido** já está associado a outro utilizador.
3. Se o email já estiver associado ao registo de outro utilizador então o sistema deve informar ao gestor que esse **email já tem um utilizador associado**.
4. Caso os dados introduzidos **colidam** com um utilizador já registado, o sistema pode **recomeçar o processo** de criar utilizador.
5. Caso os dados introduzidos não colidam com nenhum utilizador já registado, então o novo **utilizador deve ser registrado**.

2 Iniciar sessão

Requisitos de utilizador: O utilizador deve poder **realizar uma autenticação** no sistema. Esta autenticação deve ser feita de forma segura e eficiente. Para além de ser intuitiva, deve incluir uma verificação sobre os dados que o utilizador poderá inserir. Após a finalização do processo de autenticação, o utilizador deve passar a ter acesso às funções do sistema.

Requisitos de sistema:

1. O sistema deve solicitar os **dados** necessários à **autenticação** do utilizador (email e palavra-passe).
2. O sistema deve **verificar se os dados introduzidos** correspondem a algum utilizador registado.
3. Caso os dados introduzidos sejam incorretos o sistema deve **informar o utilizador** que os dados introduzidos **não correspondem** a um registo do sistema e pode **recomeçar o processo** de autenticação.

4. Caso os dados introduzidos correspondam a algum utilizador, o sistema deve **permitir ao utilizador acessar** as suas outras funções , incluindo "Criar novo utilizador" no caso de se tratar do registo do gestor.

3 Terminar sessão

Requisitos de utilizador: O utilizador deve poder **terminar a sua sessão** de forma simples e rápida. Ao terminar a sessão nenhuma informação da sessão anterior deverá manter-se disponível ao utilizador. Quando o processo termina, o sistema deve regressar a um **menu de autenticação**.

Requisitos de sistema:

1. O sistema deve **apresentar a opção** de **terminar sessão**.
2. Caso o utilizador **selecione** esta opção, então o sistema deve **retornar** a um **menu de autenticação** onde poderá apenas "Iniciar sessão".

4 Criar novo componente

Fonte: Jorge Pereira durante a sua entrevista.

Motivo: Jorge Pereira enfatizou o benefício de facilitar a **inclusão** de novos **componentes** e produtos no sistema **sem depender** de um departamento informático.

Requisitos de utilizador: O sistema deve permitir que o utilizador **introduza novos componentes no catálogo** e, assim, possibilitar que estes sejam usados na composição de produtos. O sistema deve fornecer um "formulário" ao utilizador, de modo a que este consiga facilmente perceber aquilo que descreve um componente e o que deve introduzir em cada um dos campos a preencher. Além disso, o sistema deverá verificar as informações introduzidas, garantindo a sua correção e que o peso e custo não excedem 999 quilogramas ou euros respetivamente.

Requisitos de sistema:

1. O sistema deve **solicitar** ao utilizador que insira os **dados** relativos ao **novo componente** (nome,peso,tipo de componente,custo e uma descrição) sob forma de formulário.
2. O sistema deve **recolher os dados inseridos**, **verificar** que todos são compatíveis com os tipos de dados esperados e **atribui-los** ao novo componente.
3. Após o registo dos dados o sistema deve **atribuir um identificador único** ao novo componente.
4. O sistema deve **incluir** o **novo componente** no catálogo de componentes.

5 Criar novo produto

Fonte: Jorge Pereira durante a sua entrevista.

Motivo: Jorge Pereira enfatizou o benefício de facilitar a **inclusão** de novos componentes e **produtos** no sistema sem depender de um departamento informático.

Requisitos de utilizador: O utilizador deve poder **introduzir novos produtos** no catálogo de produtos. Esta introdução deve ser realizada via um "formulário" que auxiliará os utilizadores a estar conscientes daquilo que compõe um produto. Neste "formulário" deverá existir uma lista de componentes que poderão ser selecionados para formar o produto. Este processo deve ser verificado pelo sistema, garantindo que o utilizador não pode finalizar a criação do produto com erros, por exemplo um peso e/ou custo que excedam 999 quilogramas ou euros respetivamente.

Requisitos de sistema:

1. O sistema deve **pedir** ao utilizador que insira **dados** relativos ao **produto** que deseja introduzir (nome, descrição) e atribui-los ao mesmo.
2. O sistema deve **recolher os dados inseridos**, **verificar** que todos são compatíveis com os tipos de dados esperados e **atribui-los** ao novo produto.
3. O sistema deve pedir ao utilizador para **selecionar os componentes** que vão compor o novo produto, identificando, através dos seus identificadores únicos, todos eles no catálogo de componentes e atribui-los ao produto.
4. O sistema deve **atribuir um identificador** único ao **produto** e **adicioná-lo ao catálogo** de produtos.

6 Abastecer stock de um componente

Fonte: "Software" conceptualmente similar e Inácio.

Motivo: Inácio havia mencionado que o sistema **não ia lidar** com as **encomendas** de **componentes** diretamente, ainda assim, temos que refletir o aumento de stock dos componentes e então adotamos uma tática de "software" similar e decidimos incorporar um aumento de stock manual.

Requisitos de utilizador: O utilizador deve poder **pedir um abastecimento de stock** de qualquer componente. Para isto o utilizador deverá ser questionado sobre o número de unidades de componente que deseja abastecer e o sistema deverá verificar se a quantidade introduzida é ou não válida e atualizar o stock correspondente.

Requisitos de sistema:

1. O sistema deve pedir ao utilizador a **quantidade de stock a encomendar**.
2. Se o valor for **irrealista**, o sistema deve **pedir um novo valor**.

3. Se, por algum motivo, o sistema não conseguir aumentar o stock, deve **notificar o utilizador** que o stock **não foi abastecido** e pode recomeçar o processo de abastecimento.
4. O sistema deve **aumentar** a quantidade de **stock** disponível.

7 Produzir produto

Requisitos de utilizador: O utilizador deve poder iniciar a **produção de um produto** ao especificar o identificador do produto e a quantidade que desejar produzir. O sistema deverá validar a operação antes de a confirmar. O stock dos componentes utilizados na produção do produto deve diminuir relativamente ao número de unidades do produto a ser produzidas.

Requisitos de sistema:

1. O sistema deve solicitar ao utilizador que introduza o **identificador único do produto** que **deseja produzir**.
2. O sistema deve solicitar ao utilizador que introduza a **quantidade de unidades** que deseja produzir.
3. O sistema deve **verificar** se existem **componentes suficientes** para garantir a produção.
4. Caso haja uma **falta de componentes** o sistema deve **notificar** o utilizador sobre os componentes que estão em falta.
5. Caso existam componentes suficientes então o sistema deve **pedir** que o utilizador **confirme o pedido**, revelando a quantidade a produzir e os componentes que serão utilizado.
6. Caso o utilizador recuse a confirmação o sistema pode **recomeçar o processo** de produção de produto.
7. Caso o utilizador confirme a ação, o sistema deve **iniciar a produção do produto** e **atualizar os stocks** dos componentes a ser usados.

8 Visualizar linha de montagem

Requisitos de utilizador: O utilizador deve poder **visualizar o estado** dos **produtos** na linha de montagem e quais os componentes já instalados no produto, que deverá estar identificado por um identificador único e o seu nome. O sistema deve apresentar uma "fila de espera" que contém os produtos que aguardam para ser produzidos. A linha de montagem suporta a produção concorrente de 3 produtos(cadeiras).

Requisitos de sistema:

1. Caso a linha de montagem se encontre vazia, então o sistema deve **notificar o utilizador** que não existem produtos em produção.

2. O sistema deve **apresentar os produtos** em produção com informação relevante sobre eles (identificador único, nome), os componentes já instalados e a instalar e uma estimativa do tempo necessário para terminar a sua montagem bem como o tempo decorrido desde o inicio da montagem.
3. O sistema deve apresentar a **fila de espera** dos produtos com o nome do produto e quantidade a produzir.

9 Visualizar catálogo de componentes

Fonte: Ex-funcionários de fábricas de cadeiras.

Motivo: Foi realçado o benefício de poder **verificar** facilmente **informação** acerca dos **componentes** e produtos relevantes na fábrica, desta forma, decidimos implementar a capacidade de visualizar os catálogos de componentes e produtos.

Requisitos de utilizador: O **utilizador** deve poder **aceder** ao **catálogo de componentes**, onde são apresentadas informações essenciais de cada componente e o stock disponível. Esta funcionalidade deve assegurar que o utilizador possa consultar dados como identificador único, nome, peso, tipo e custo de cada componente, de forma clara e comprehensiva.

Requisitos de sistema:

1. Caso o catálogo de componentes se encontre vazio, o sistema deve **notificar o utilizador** que a lista se encontra vazia.
2. O sistema deve **apresentar uma lista de componentes** com dados essenciais dos mesmos (identificador único, nome, peso, tipo e custo) e a sua quantidade em stock.
3. O sistema deve permitir ler a **descrição** dos **componentes**.

10 Visualizar histórico de produção

Fonte: Análise ao vivo do ambiente de trabalho de uma fábrica ativa.

Motivo: Nas nossas análises notamos a utilidade de um histórico de produção para os **funcionários**. Esta função é útil no dia a dia dos mesmos e parece **motivá-los** ao facilitar o reconhecimento do esforço e progresso realizados.

Requisitos de utilizador: O sistema deve permitir que o utilizador **aceda ao histórico** de produção de forma organizada e informativa, exibindo todos os **produtos** já produzidos, acompanhados de informações detalhadas como identificador único, nome, custo e peso. Além disso, o sistema deve fornecer **estatísticas úteis** sobre cada produto como o tempo de produção médio, quantidade total produzida e o custo de quantidade total produzida.

Requisitos de sistema:

1. O sistema deve **apresentar uma lista** com todos os **produtos já produzidos**, apresen-

tando, para cada produto, o seu identificador único, nome, custo (custo dos componentes) e peso.

2. O sistema deve **apresentar estatísticas** sobre cada produto como tempo de produção médio, quantidade total produzida e custo de quantidade total produzida.

11 Remover produto do catálogo de produtos

Fonte: Jorge Pereira durante a sua entrevista.

Motivo: O gerente sénior fez-nos entender a relevância de poder remover produtos do catálogo, uma vez que, para qualquer produtora de cadeiras a longo prazo, o **descontinuar** de componentes e **substituição** de produtos por novas versões é expectável e, eventualmente, **inevitável**.

Requisitos de utilizador: O sistema deve **permitir** que se **remova produtos do catálogo** para que seja **impossível** iniciar a **produção** deles **posteriormente**. Para tal, o utilizador deve ser guiado na identificação do produto e deve receber "feedback" claro em caso de erros. O sistema deve solicitar a confirmação do utilizador antes de concluir a remoção, apresentando os dados do produto para revisão. Os produtos em produção não devem ser afetados, apenas deve ser afetada a capacidade de produzir e de ver o produto no catálogo de produtos.

Requisitos de sistema:

1. O sistema deve pedir ao utilizador que introduza o **identificador único** do **produto** que deseja remover.
2. Caso o sistema não identifique o produto deve informar ao utilizador que **não é possível remover o produto** do catálogo de produtos.
3. Caso esse produto exista, o sistema deve pedir que o utilizador **confirme o pedido**, revelando o produto a remover do catálogo com o seu identificador único e nome.
4. Caso o utilizador recuse a confirmação o sistema pode **recomeçar o processo de remoção** de produto.
5. Caso o utilizador confirme a ação, o sistema deve **remover o devido produto** do catálogo de produtos.

12 Diminuir stock de produto

Fonte: Inácio e "Software" conceptualmente similar.

Motivo: Inácio havia mencionado que o sistema **não ia lidar** com as **vendas** de **produto** diretamente, ainda assim, temos que **refletir** a diminuição de stock, quer por **vendas**, quer por **danos inesperados**, no sistema de modo a distinguir a demanda de produção de cada produto. Com isto e através da análise de outros "Software" especificamos este requisito.

Requisitos de utilizador: O sistema deve permitir que o utilizador **reduza o stock** de produtos de maneira **prática e precisa**. O processo deve iniciar com um pedido do identificador único do produto e da quantidade a ser retirada, com validações para garantir que a quantidade inserida não ultrapasse o stock disponível. O utilizador deve ser informado dos impactos da operação, com o stock atual e o stock resultante, e deve ter de confirmar a ação. O sistema deve atualizar o stock de forma consistente, garantindo o correto acompanhamento do inventário.

Requisitos de sistema:

1. O sistema deve solicitar ao utilizador que introduza o **identificador único do produto** cujo stock pretende diminuir.
2. Se o produto não existir, o sistema deve **notificar o utilizador que não foi possível** reduzir o stock e pode **recomeçar o processo** de diminuição de stock.
3. Caso esse produto exista, o sistema deve perguntar ao utilizador qual a **quantidade a ser retirada** do stock.
4. Caso a quantidade seja superior à disponível em stock deve informar o utilizador que não é possível remover a quantidade especificada do stock do produto.
5. O sistema deve pedir que o utilizador **confirme o pedido**, revelando o produto cujo stock vai diminuir com o seu identificador único, nome , stock atual e o stock após realizar a operação.
6. Caso o utilizador recuse a confirmação, o sistema pode **recomeçar o processo** de diminuição de stock.
7. Caso o utilizador confirme a ação, o sistema deve **diminuir o stock** do devido produto.

13 Diminuir stock de um componente

Fonte: Ex-funcionários de fábricas de cadeiras.

Motivo: Alguns ex-funcionários realçaram a importância de um método de **diminuição de stock** de componente **direto**. Estávamos sob a impressão que o stock de um componente diminuiria ao produzir um produto e que isso seria suficiente, mas, **perdas inesperadas de stock** podem acontecer, por exemplo, com um desastre natural como um terramoto que arruine parte do stock, logo decidimos especificar este requisito.

Requisitos de utilizador: O utilizador deve poder **diminuir o stock de um componente** para refletir percas de stock inesperadas. Para isto o utilizador deve introduzir o identificador único do componente, seguido pela quantidade que deseja retirar, com validações para prevenir erros. O utilizador deve ser informado sobre o impacto da operação no sistema antes de se dar por completa e esta deve solicitar uma confirmação final ao utilizador.

Requisitos de sistema:

1. O sistema deve solicitar ao utilizador que introduza o **identificador único do compo-**

nente cujo stock pretende diminuir.

2. Caso esse componente exista, o sistema deve perguntar ao utilizador qual a **quantidade a ser retirada** do stock.
3. Se o componente não existir ou a quantidade for superior à disponível em stock, o sistema deve **notificar o utilizador que não foi possível** reduzir o stock e pode **recomeçar o processo** de diminuição de stock.
4. Caso o componente exista e a sua quantidade não seja superior à disponível em stock, o sistema deve pedir que o utilizador **confirme o pedido**, revelando o componente cujo stock vai diminuir com o seu identificador único, nome, stock atual e o stock após realizar a operação.
5. Caso o utilizador recuse a confirmação, o sistema pode **recomeçar o processo** de diminuição de stock.
6. Caso o utilizador confirme a ação, o sistema deve **diminuir o stock** do devido componente.

14 Visualizar catálogo de produtos

Fonte: Ex-funcionários de fábricas de cadeiras.

Motivo: Foi realçado o benefício de poder **verificar** facilmente **informação** acerca dos componentes e **produtos** relevantes na fábrica, desta forma, decidimos implementar a capacidade de visualizar os catálogos de componentes e produtos.

Requisitos de utilizador: O sistema deve proporcionar ao utilizador a **possibilidade de consultar**, de forma clara e organizada, o **catálogo de produtos**. Esta funcionalidade deve exibir uma lista completa com informações relevantes para cada produto, incluindo identificador único, nome, peso, custo e a quantidade disponível em stock.

Requisitos de sistema:

1. Caso o catálogo de produtos se encontre vazio, então o sistema deve **notificar o utilizador** que a lista se encontra vazia.
2. O sistema deve apresentar uma **lista de produtos** com dados essenciais dos mesmos (identificador único, nome, peso e custo) e quantidade em stock.
3. O sistema deve permitir ler a **descrição** dos **produtos**.

2.3 Requisitos não funcionais

Tempos de execução

- Todas as interações do sistema com o utilizador devem ser processadas em **menos de 5 segundos**, com um tempo de resposta **ideal inferior a 1 segundo**.

Ambiente de execução do programa

- O sistema deve correr em um ambiente "**Microsoft Windows**".

2.4 Validação dos requisitos estabelecidos

Para verificar que os requisitos previamente estabelecidos iam de encontro com aquilo pretendido, **garantimos** que o nosso **cliente detinha toda a informação** relevante para tomar a decisão de prosseguir com o projeto.

Foi realizada uma reunião com Inácio, com o intuito de assegurar que os requisitos propostos se **alinjavam com a sua visão para a linha de montagem**. Primeiro verificamos que todas as **funcionalidades** propostas por Inácio foram **devidamente expressas** nos requisitos e que todas as outras correspondiam a funcionalidades que **pretendia incorporar**.

A seguir, **discutimos custos e tempos** necessários ao **desenvolvimento** do "software" delineado. Foi confirmado que os custos bem como o tempo de produção associados ao desenvolvimento do "software" se encontravam nos limites definidos por Inácio e, assim, tomamos os requisitos levantados como válidos.

3 Especificação e modelação do "software"

Neste capítulo temos como objetivo **especificar** de forma palpável e útil, tanto ao nível do desenvolvimento interno como para auxiliar na exposição do projeto aos futuros utilizadores, os nossos **requisitos funcionais**.

3.1 Apresentação geral da especificação

Visamos descrever não só o **comportamento** do sistema como também uma **estrutura** adequada que garanta a realização dos requisitos. Decidimos adotar representações da linguagem **UML** ("Unified Modeling Language"); consideramo-la competente e, além disso, é familiar a todos os membros da equipa de desenvolvimento.

Para demonstrar a **estrutura** optámos por utilizar um **modelo de domínio**. Este modelo permite-nos **manter** um certo **nível de liberdade** no futuro do projeto quando comparado, por exemplo, a um diagrama de classes, o que achamos oportuno dada a nossa falta de conhecimento no domínio da aplicação. A **simplicidade** desta representação **facilita a troca de ideias** com os diversos funcionários experientes (no fabrico de cadeiras) com os quais pretendemos discutir e avaliar a qualidade das nossas propostas, ajudando assim na identificação de possíveis defeitos nelas.

Outro benefício deste modelo é o facto de **promover** um **desenvolvimento orientado a objetos**; ele permitir-nos-á criar classes que correspondem diretamente a entidades no modelo definido sistematicamente, proporcionando uma **diminuição no tempo de desenvolvimento** bem como garantindo que o desenvolvimento não se afasta da visão previamente definida. Essencialmente, a escolha que fizemos garante-nos uma "**pré-produção**" relativamente ao **código** por escrever.

Para modelar o **comportamento** decidimos utilizar dois diagramas:

- **Diagrama de Use Case**
- **Diagrama de Atividades**

Estes dois diagramas foram escolhidos porque se complementam muito bem e, tal como o modelo de domínio, contribuem para um grau de complexidade acessível aos nossos assessores.

O diagrama de **Use Case** ilustra a interação entre o utilizador e o sistema e os diferentes papéis dos utilizadores. Já o diagrama de **atividades** dá uma ideia de como o fluxo do programa ou seja, que tipo de operações serão possíveis num estado particular do sistema, como estas se encadeiam entre si e as re oferecendo um excelente apoio para o diagrama de **Use Case**.

3.2 Aspectos estruturais

Como referido na secção anterior, construímos um **modelo de domínio**. O modelo é composto pelas seguintes entidades:

1. **Produto**: Representa todas as cadeiras que já foram inseridas pelos utilizadores no nosso sistema. Esta entidade é caracterizada por diversos atributos como **Identificador**, **Nome**, **Descrição**, **Stock**, **Catálogo**, **Custo** e **Peso**.
2. **Componente**: Representa todos os componentes que podem ser usados para produzir cadeiras. Esta entidade é composta pelos **mesmos atributos** da entidade **Produto**, com acréscimo do **Tipo de Componente**.
3. **Instância de Produto**: Permite a distinção entre as cadeiras fabricadas através da atribuição de um identificador; serve para distinguir cadeiras individuais que são unidades do mesmo "**Produto**".
4. **Linha de montagem**: Representa o agrupamento de instâncias de produto que estão a ser produzidas.
5. **Utilizador**: Representa todos os utilizadores da plataforma, sendo caracterizada pelos atributos "**E-mail**", **Palavra-Passe** e **Administrador**.

É também importante notar estas quatro relações:

- **Produto – Componente**: É fundamental para representar a composição dos produtos do sistema, já que os produtos são compostos por componentes;
- **Utilizador – Componente**: Representa a possibilidade dos utilizadores serem capazes de inserir novos componentes e abastecerem o stock dos já existentes;
- **Utilizador – Produto**: Representa a possibilidade do utilizador ser capaz de criar novos produtos, que posteriormente poderão ser produzidos;
- **Utilizador – Instância de Produto**: Permite ao utilizador fabricar novos produtos.

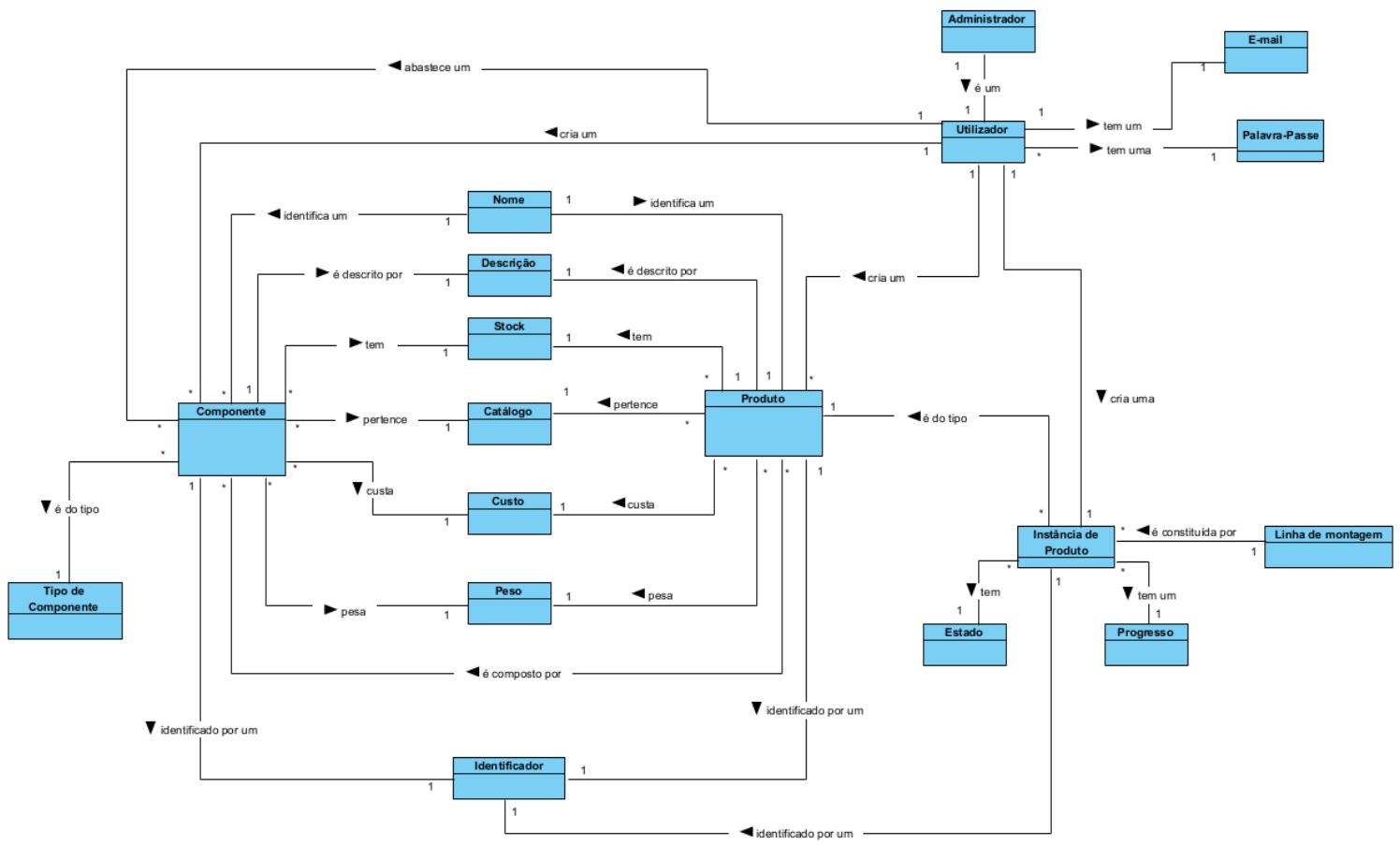


Figura 3.1: Modelo de Domínio

Caracterização das entidades e atributos:

- **Utilizador:** Caracteriza os diferentes utilizadores que podem interagir com o sistema;
- **Administrador:** Tipo de utilizador que possui todas as permissões de um utilizador comum, com a adição de que pode criar novos utilizadores;
- **E-mail:** Corresponde ao e-mail de um certo utilizador;
- **Palavra-Passe:** Corresponde à palavra-passe de certo(s) utilizador(es);
- **Componente:** Caracteriza os diferentes componentes que existem no nosso sistema;
- **Produto:** Caracteriza os diferentes produtos que existem no nosso sistema;
- **Instância de Produto:** Serve para distinguir produtos individuais na linha de montagem, garantindo que não sejam apenas um número no stock, podendo assim serem diferenciados;
- **Nome:** Representa o nome de certo componente ou produto;
- **Descrição:** Caracteriza fisicamente e/ou funcionalmente certo componente ou produto;
- **Stock:** Número que representa as unidades em stock de certo componente ou produto;
- **Catálogo:** Representa uma coleção de produtos ou componentes de um certo tipo, excluindo aqueles que estão descontinuados;
- **Custo:** Valor em euros do custo de produção/compra de certo produto/componente;
- **Peso:** Valor em quilogramas do peso associado a certo produto ou componente;
- **Identificador:** Número que identifica um certo componente, produto ou instância de produto. Cada entidade tem o seu próprio conjunto de identificadores, por exemplo, tanto a entidade componente como a entidade produto, podem ter um identificador com o valor de 1;
- **Estado:** Representa o estado relativo à linha de montagem de cada instância de produto. "E" - Significa que está na fila de espera para ser produzido; "C" - Significa que já foi produzido; "P" - Significa que está a ser produzido;
- **Progresso:** Percentagem aproximada que representa o progresso da produção de uma certa instância de produto;
- **Linha de montagem:** É caracterizada pelo conjunto das instâncias de produtos que estão a ser produzidas;
- **Tipo de componente:** Refere-se à peça que está a ser introduzida. Este campo pode, por exemplo, indicar que a peça é uma "roda" ou um "braço direito" de uma cadeira.

3.3 Aspectos comportamentais

Relativamente aos aspectos comportamentais, começamos por elaborar os **diagramas de Use Case** que representam as interações dos atores no sistema. A seguir, **detalhamos cada um** dos Use Case representados nos diagramas para fornecer uma visão clara dos processos que os definem, descrevendo como os atores e o sistema se relacionam de forma a alcançar o resultado esperado. Por fim, **criamos um diagrama de atividades** para facilitar a percepção do fluxo das ações realizadas pelos utilizadores no sistema.

3.3.1 Diagramas de Use Case

Para organizar os Use Case, decidimos construir três diagramas, cada um referente a um certo (sub)sistema:

- Diagrama referente ao **sistema geral (diagrama Use Case do Sistema)**, ou seja, todas as funcionalidades a que o utilizador e administrador têm acesso numa forma geral;
- Diagrama referente à "**Gestão de Componentes**" (**diagrama Use Case do subsistema de componentes**), em que são representadas todas as funcionalidades que podem ser executadas relativamente aos componentes (entidade Componente no modelo de domínio);
- Diagrama referente à "**Gestão de Produtos**" (**diagrama Use Case do subsistema de produtos**), que refere todas as funcionalidades permitidas relativamente aos produtos e/ou instância de produtos (entidade Produto e Instância de Produto no modelo de domínio).

É de realçar que, nos subsistemas de componentes e de produtos, o ator "**Administrador**" está implícito no ator "Utilizador", visto que, ambos são capazes de utilizar todas as funcionalidades disponíveis.

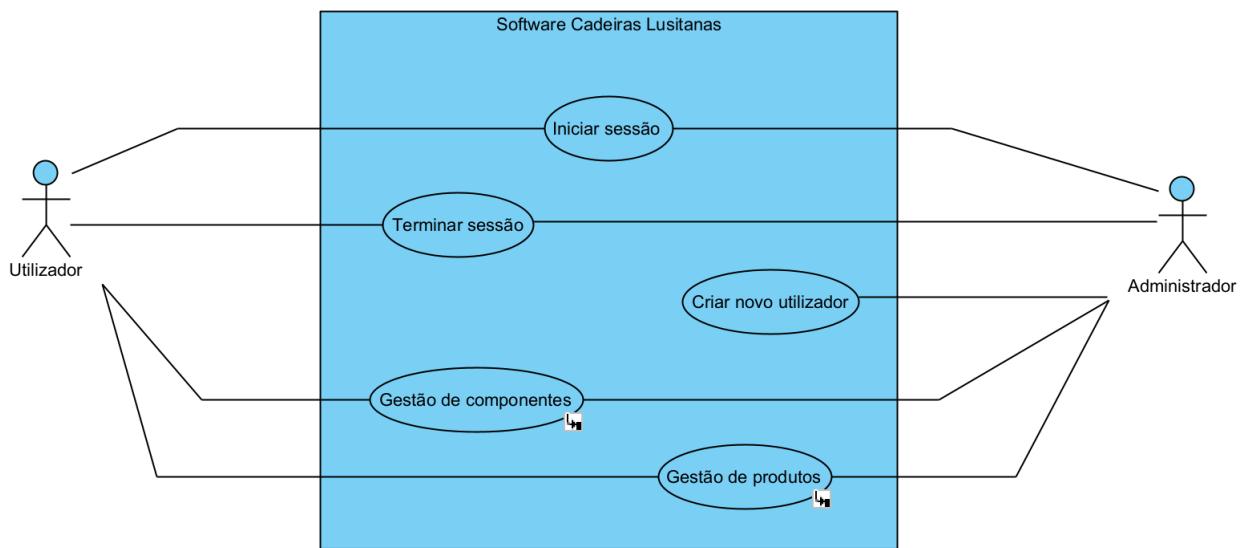


Figura 3.2: Diagrama de Use Case do Sistema

Proveniente dos requisitos enunciados na subsecção 2.2, surgiram diversos Use Case com o intuito de definir as funcionalidades do sistema:

1. **Iniciar sessão**: Responsável por iniciar sessão de um ator no sistema (requisito 2);
2. **Terminar sessão**: Responsável por terminar sessão de um ator no sistema (requisito 3);
3. **Criar novo utilizador**: Permite ao ator Administrador criar novas contas de utilizadores (requisito 1).

Como é de notar no diagrama, temos ainda a "**Gestão de Componentes**" e a "**Gestão de Produtos**" que, apesar de não serem Use Case, são subsistemas compostos por Use Case referentes a certas entidades e, como disponibilizam funcionalidades aos atores, devem ser expressos no diagrama do sistema.

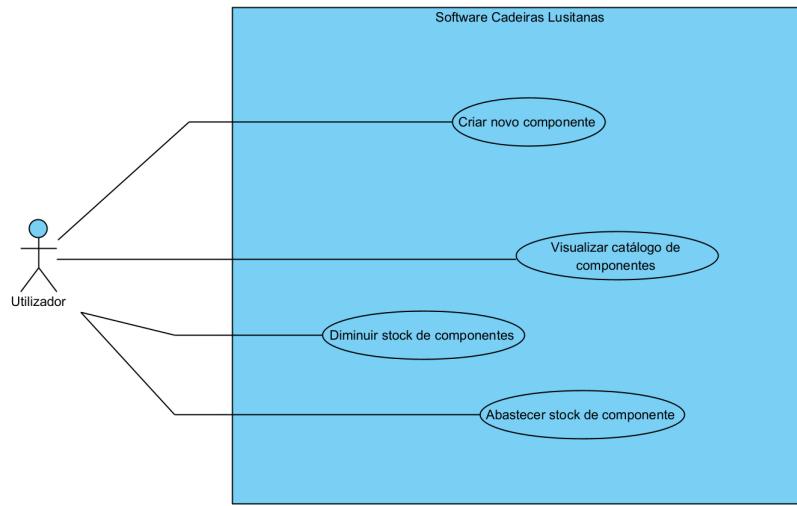


Figura 3.3: Diagrama de Use Case do subsistema de componentes.

Neste diagrama serão apresentados os seguintes Use Case relacionados com a gestão e manutenção de componentes que foram gerados através dos requisitos mencionados previamente:

1. **Criar novo componente:** Permite ao ator criar um novo componente que será inserido no catálogo de componentes (requisito 4);
2. **Visualizar catálogo de componentes:** Apresenta os componentes disponíveis no sistema (requisito 9);
3. **Diminuir stock de componentes:** Caso possível, diminui uma quantidade do stock de um certo componente (requisito 13);
4. **Abastecer stock de componente:** Caso possível, aumenta uma quantidade do stock de um certo componente (requisito 6).

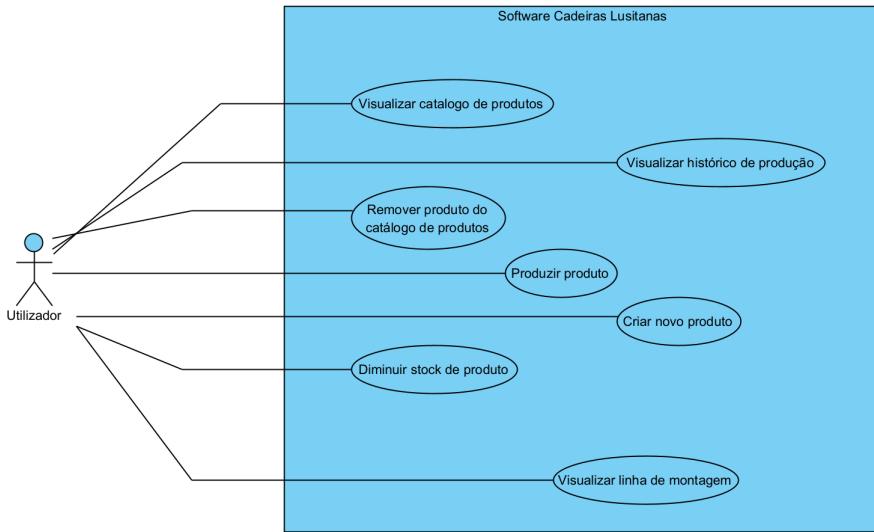


Figura 3.4: Diagrama de Use Case do subsistema de produtos

De forma idêntica ao diagrama do subsistema de componentes, este diagrama apresenta-nos as funcionalidades do sistema, relativas à gestão de produtos, descritas pelos seguintes Use Case:

1. **Visualizar catálogo de produtos:** Apresenta os produtos que estão no catálogo de produtos, ou seja, os que podem ser produzidos (requisito 14);
2. **Visualizar histórico de produção:** Apresenta um histórico dos últimos produtos que foram produzidos e algumas estatísticas(requisito 10);
3. **Remover produto do catálogo de produtos:** Permite ao ator remover um certo produto do catálogo de produtos (requisito 11);
4. **Producir produto:** Responsável por iniciar a produção de um certo produto requisitado pelo ator, aumentando stock correspondente(requisito 7);
5. **Criar novo produto:** Permite ao ator criar um novo produto composto pelos componentes escolhidos. Caso seja um produto válido, este será inserido no catálogo de produtos (requisito 5);
6. **Diminuir stock de produto:** Caso possível, diminui uma quantidade do stock de um certo produto (requisito 12);
7. **Visualizar linha de montagem:** Apresenta ao ator os produtos que estão a ser produzidos (requisito 8);

3.3.2 Use Case

Na presente secção iremos enumerar as funcionalidades do sistema para a execução de cada um dos Use Case, referindo os requisitos que lhes deram origem.

USE CASE 1: Criar novo utilizador. (Referente ao requisito 1)

Atores: Administrador.

Descrição: Ator cria um novo registo de utilizador.

Pré-condição: Ator está autenticado e é um Administrador.

Pós-condição: Novo utilizador adicionado ao sistema ou operação cancelada.

Fluxo Normal:

1. O ator insere o email do novo utilizador.
2. O sistema valida o email introduzido.
3. O ator insere a palavra-passe que quer associar ao novo utilizador.
4. O sistema regista o novo utilizador.

Fluxo de Exceção 1: [Email já registado no sistema] (passo 2)

- 2.1 O sistema informa que já existe um utilizador registado com o email introduzido.

USE CASE 2: Iniciar sessão. (Referente ao requisito 2)

Atores: Utilizador.

Descrição: Ator inicia sessão no "software" da "Cadeiras Lusitanas".

Pré-condição: Ator não está autenticado.

Pós-condição: Ator fica autenticado.

Fluxo Normal:

1. O ator indica o seu email e senha.
2. O sistema valida o email e senha introduzidos.
3. O sistema apresenta o menu da "Cadeiras Lusitanas".

Fluxo de Exceção 1: [Validação de email e senha falhou] (passo 2)

- 2.1 O sistema informa que o email e senha introduzidos não são válidos.

USE CASE 3: Terminar sessão. (Referente ao requisito 3)

Atores: Utilizador.

Descrição: Ator termina uma sessão no "software" da "Cadeiras Lusitanas".

Pré-condição: Ator está autenticado.

Pós-condição: Ator deixa de estar autenticado.

Fluxo Normal:

1. O ator carrega no botão para terminar sessão.
2. O sistema termina a sessão corretamente.

USE CASE 4: Criar novo componente. (Referente ao requisito 4)

Atores: Utilizador.

Descrição: Ator adiciona um novo componente ao catálogo de componentes.

Pré-condição: Ator está autenticado.

Pós-condição: Novo componente adicionado ao sistema ou operação cancelada.

Fluxo Normal:

1. O sistema solicita ao ator que insira o nome, peso, tipo de componente, custo e uma descrição relativos ao novo componente.
2. O sistema atribui um identificador único ao novo componente.
3. O sistema regista o novo componente na base de dados.

Fluxo de Exceção 1: [Erro na inserção de dados] (passo 1)

- 1.1 O sistema informa o ator que não foi possível adicionar o novo componente.

USE CASE 5: Criar novo produto. (Referente ao requisito 5)

Atores: Utilizador.

Descrição: Ator adiciona um novo produto ao catálogo de produtos.

Pré-condição: Ator está autenticado e existem, no sistema, componentes necessários à produção do produto.

Pós-condição: Novo produto adicionado ao catálogo do sistema ou operação cancelada.

Fluxo Normal:

1. O sistema solicita ao ator que insira o nome e descrição relativos ao novo produto.
2. O ator seleciona de um menu os componentes que vão compor o novo produto.(se o produto usufruir da mesma peça x vezes deve ser selecionado x vezes)

3. O sistema atribui um identificador único ao novo produto.
4. O sistema regista o novo produto na base de dados.

Fluxo de Exceção 1 [Erro na inserção de dados] (passo 1)

- 1.1 O sistema informa o ator que não foi possível adicionar o novo produto.

USE CASE 6: Abastecer stock de componente. (**Referente ao requisito 6**)

Atores: Utilizador.

Descrição: Ator abastece o stock de um dado componente.

Pré-condição: Ator está autenticado e existe pelo menos um componente no sistema.

Pós-condição: Stock do componente abastecido ou operação cancelada.

Fluxo Normal:

1. O ator seleciona o componente cujo stock deseja aumentar.
2. O ator indica a quantidade que deseja adicionar ao stock do componente.
3. O sistema valida quantidade.
4. O sistema aumenta o stock do componente.
5. O sistema notifica ator que o stock foi aumentado com sucesso.

Fluxo de Exceção 1: [Quantidade inválida] (passo 3)

- 3.1 O sistema notifica o ator que quantidade introduzida é inválida.

Fluxo de Exceção 2: [Erro a aumentar stock] (passo 4)

- 4.1 O sistema notifica o ator que ocorreu um erro ao aumentar o stock.

USE CASE 7: Produzir produto. (**Referente ao requisito 7**)

Atores: Utilizador.

Descrição: Este Use Case descreve o processo pelo qual um utilizador solicita a produção de um determinado produto, especificando a quantidade desejada. O sistema valida a viabilidade da produção, verificando a disponibilidade dos componentes necessários e atualizando o stock após a conclusão da produção.

Pré-condição: O ator está autenticado.

Pós-condição: Sistema começa a produção dos produtos e o stock associado está atualizado ou operação cancelada.

Fluxo Normal:

1. O ator insere o identificador do produto que deseja produzir.
2. O sistema valida o identificador.
3. O ator insere a quantidade de produto que deseja produzir.
4. O sistema verifica se existem componentes suficientes para a produção do produto.
5. O sistema valida a quantidade a produzir.
6. O sistema pede ao ator que confirme o pedido.
7. O sistema atualiza o stock de componentes.
8. O sistema inicia a produção dos produtos e atualiza o seu stock gradualmente.

Fluxo de Exceção 1 [Identificador não existe] (passo 1)

- 1.1 O sistema informa o ator que o identificador de produto não está associado a qualquer produto.

Fluxo de Exceção 2: [Quantidade de componentes insuficiente] (passo 4)

- 4.1 O sistema notifica o ator de que não existem componentes suficientes em stock para produzir a quantidade de produtos introduzida.

Fluxo Alternativo 1: [Ator não permitiu a operação] (passo 6)

- 6.1 O sistema cancela a operação.

USE CASE 8: Visualizar linha de montagem. (**Referente ao requisito 8**)

Atores: Utilizador.

Descrição: Este Use Case descreve o processo pelo qual o utilizador visualiza a linha de montagem, verificando detalhes dos produtos em produção.

Pré-condição: Ator está autenticado.

Pós-condição: Ator visualiza a informação atualizada da linha de montagem.

Fluxo Normal:

1. O sistema verifica a linha de montagem.
2. O sistema apresenta ao ator a fila de produtos em produção, especificando o identificador, nome, os componentes já instalados e a instalar e uma estimativa do tempo necessário para terminar a sua montagem bem como o tempo decorrido desde o inicio da montagem.

- O sistema deve apresentar a fila de espera de produtos com o seu nome e quantidade a produzir.

Fluxo de Exceção 1 [A fila de produção está vazia] (passo 1)

- O sistema informa o ator de que não existem produtos em produção.

USE CASE 9: Visualizar catálogo de componentes. (**Referente ao requisito 9**)

Atores: Utilizador.

Descrição: Ator visualiza o catálogo de componentes.

Pré-condição: Ator está autenticado.

Pós-condição: Sistema apresenta o catálogo de componentes.

Fluxo Normal:

- O sistema obtém o catálogo de componentes através da base de dados.
- O sistema apresenta o catálogo de componentes ao ator, especificando para cada um o seu identificador único, nome, peso, tipo, custo e descrição.

Fluxo de Exceção 1 [Catálogo de componentes vazio] (passo 1)

- O sistema informa o ator de que não existem componentes no catálogo.

USE CASE 10: Visualizar histórico de produção. (**Referente ao requisito 10**)

Atores: Utilizador.

Descrição: Este Use Case caracteriza o processo no qual o utilizador vê o histórico de produção por meio de uma lista dos produtos produzidos anteriormente. Além disso, este histórico apresenta estatísticas de produção agregadas por tipo de produto.

Pré-condição: Ator está autenticado.

Pós-condição: O utilizador visualiza a lista de produtos já produzidos, com detalhes sobre cada um e estatísticas agregadas sobre o histórico de produção.

Fluxo Normal:

- O sistema obtém o histórico de produção através da base de dados.
- O sistema apresenta uma lista com todos os produtos já produzidos, especificando o identificador, nome, peso e custo de cada produto.
- O sistema apresenta um conjunto de estatísticas: tempo de produção médio, quantidade total e custo total para cada tipo de produto já produzido.

USE CASE 11: Remover produto do catálogo de produtos. (**Referente ao requisito 11**)

Atores: Utilizador.

Descrição: Ator remove um produto do catálogo de produtos, de modo a refletir a sua descontinuidade de produção.

Pré-condição: Ator está autenticado e existe pelo menos um produto no catálogo.

Pós-condição: Produto removido do sistema ou operação cancelada.

Fluxo Normal:

1. O sistema solicita ao ator que insira o identificador único do produto que deseja remover.
2. O sistema valida identificador introduzido.
3. O sistema solicita ao ator que confirme a remoção do produto.
4. O sistema atualiza o catálogo de produtos na base de dados.

Fluxo de Exceção 1: [Erro na procura do identificador de produto] (passo 2)

- 2.1 O sistema informa o ator de que não foi possível remover o produto.

Fluxo Alternativo 1: [Atores não permitiu a operação] (passo 3)

- 3.1 O sistema cancela a operação.

USE CASE 12: Diminuir stock de produto. (**Referente ao requisito 12**)

Atores: Utilizador.

Descrição: Ator diminui o stock de um dado produto, de modo a refletir escoamento de stock. **Pré-condição:** Ator está autenticado e existe pelo menos um produto em stock no sistema.

Pós-condição: Stock de produto diminui ou operação cancelada.

Fluxo Normal:

1. O ator insere o identificador único do produto cujo stock deseja diminuir.
2. O sistema valida o identificador introduzido.
3. O ator insere a quantidade a remover.
4. O sistema valida a quantidade introduzida.
5. O sistema pede ao ator para confirmar a redução, revelando o produto cujo stock vai diminuir com o seu identificador único, nome, stock atual e o stock após realizar a operação.
6. O sistema atualiza o stock do produto.

Fluxo de Exceção 1: [Identificador não existe] (passo 2)

2.1 O sistema notifica o ator que o identificador introduzido não existe no sistema.

Fluxo de Exceção 2: [Quantidade a remover inválida] (passo 4)

4.1 O sistema informa o ator de que não foi possível remover tal quantidade do stock do produto pedido.

Fluxo Alternativo 1 [Ator não permitiu a operação] (passo 5)

5.1 O sistema cancela a operação.

USE CASE 13: Diminuir stock de componentes. (**Referente ao requisito 13**)

Atores: Utilizador.

Descrição: Ator diminui o stock de um dado componente no "software" da Cadeiras Lusitanas, devido a perdas de stock inesperadas, como por exemplo componentes danificados.

Pré-condição: Ator está autenticado e existe pelo menos um componente em stock no sistema.

Pós-condição: Stock de componente diminui ou operação cancelada.

Fluxo Normal:

1. O ator insere o identificador único do componente cujo stock deseja diminuir.
2. O sistema valida o identificador introduzido.
3. O ator insere a quantidade a remover.
4. O sistema valida a quantidade introduzida.
5. O sistema pede ao ator para confirmar a redução, revelando o componente cujo stock vai diminuir com o seu identificador único, nome, stock atual e o stock após realizar a operação.
6. O sistema atualiza o stock do componente.

Fluxo de Exceção 1: [Identificador não existe] (passo 2)

2.1 O sistema notifica o ator que o identificador introduzido não existe no sistema.

Fluxo de Exceção 2: [Quantidade a remover inválida] (passo 4)

4.1 O sistema informa o ator de que não foi possível remover a quantidade introduzida do stock do componente pedido.

Fluxo Alternativo 1 [Ator não permitiu a operação] (passo 5)

5.1 O sistema cancela a operação.

USE CASE 14: Visualizar catálogo de produtos. (**Referente ao requisito 14**)

Atores: Utilizador.

Descrição: Ator visualiza o catálogo de produtos.

Pré-condição: Ator está autenticado.

Pós-condição: Catálogo de produtos é apresentada.

Fluxo Normal:

1. O sistema obtém o catálogo de produtos através da base de dados.
2. O sistema apresenta o catálogo de produtos ao ator, especificando para cada produto o seu identificador único, nome, peso, custo, quantidade em stock e descrição.

Fluxo de Exceção 1: [Catálogo de produtos vazio] (passo 1)

1.1 O sistema informa o ator de que não existem componentes no catálogo

3.3.3 Diagrama de atividades

O diagrama de atividades ilustra o fluxo de utilização do "software". O utilizador inicia o processo ao "**Iniciar sessão**" (**Use Case 2**), após o qual o sistema apresenta o **histórico de produção e as estatísticas de produção** (**Use Case 10**) e as opções.

Decidimos organizar as opções em **gestão de componentes** e **gestão de produtos** para desenvolver um fluxo intuitivo, além das opções "**Terminar sessão**" (**Use Case 3**) e, no caso do utilizador se tratar de um administrador, a opção de "**Criar novo utilizador**" (**Use Case 1**).

No diagrama de atividades a decisão "[componentes]" e a decisão "[produtos]" simbolizam, respetivamente, que o utilizador selecionou a opção "Gestão de Componentes" ou "Gestão de Produtos". Caso o façam, as várias opções relevantes e o respetivo catálogo serão apresentados (**Use Case 9 e 14**).

Na **gestão de componentes**, o utilizador tem a opção de:

- "**Criar novo componente**" (**Use Case 4**)
- "**Diminuir stock de componentes**" (**Use Case 13**)

Com o catálogo de componentes visível o sistema permite que o utilizador abasteça o stock de um componente ("**Abastecer o stock**" (**Use Case 6**)).

Na **gestão de produtos**, o utilizador tem a opção de:

- "**Criar novo produto**" (**Use Case 5**)
- "**Producir produto**" (**Use Case 7**)
- "**Visualizar linha de montagem**" (**Use Case 8**)
- "**Remover produto**" (**Use Case 11**)
- "**Diminuir stock de produto**" (**Use Case 12**)

Segue-se o diagrama de atividades descrito.

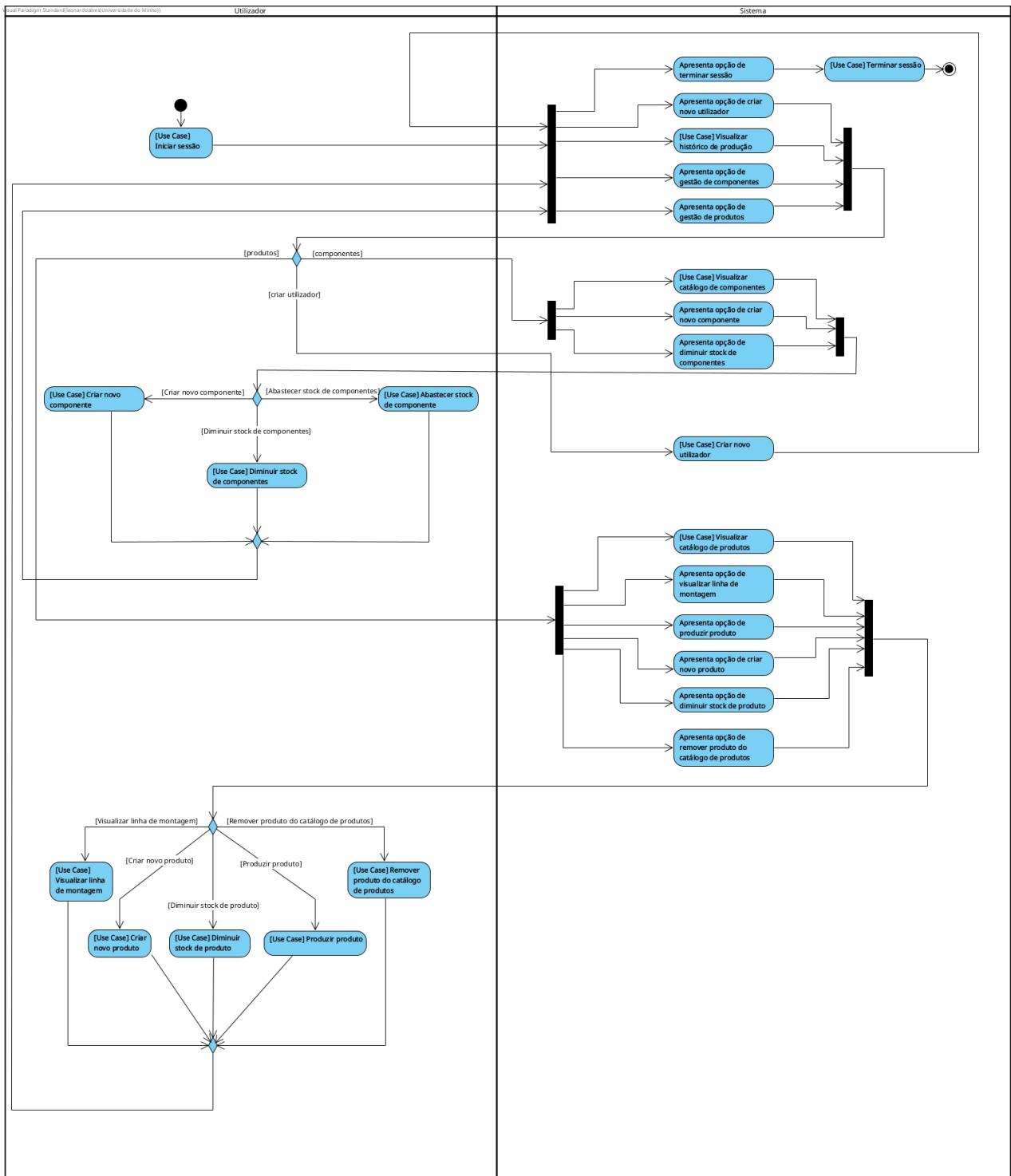


Figura 3.5: Diagrama de atividades

4 Conceção do Sistema de Dados

4.1 Apresentação geral da estrutura do sistema de dados

Com a decisão prévia de utilizar **Microsoft SQL Server** começamos o processo de desenvolvimento da base de dados pela realização de um **modelo lógico** que visa descrever as entidades e os seus relacionamentos. Para definir o modelo lógico começamos por identificar as **entidades relevantes** e criar uma tabela para cada uma. Dado o baixo número de entidades no nosso sistema, o **modelo lógico** resultante manteve-se relativamente **simples e conciso**. Em seguida **atribuímos as chaves primárias** que melhor se adequavam a cada entidade. **Adicionamos os relacionamentos**, dos quais se originou uma tabela de, proveniente de um relacionamento de cardinalidade N para N, "**Produto_tem_componente**" onde registamos o número de unidades ("Quantidade") de cada um dos componentes necessários para produzir certo produto.

Para **finalizar** colocamos todos os **atributos necessários** em todas as entidades e definimos as suas respetivas chaves estrangeiras e obtemos o seguinte esquema:

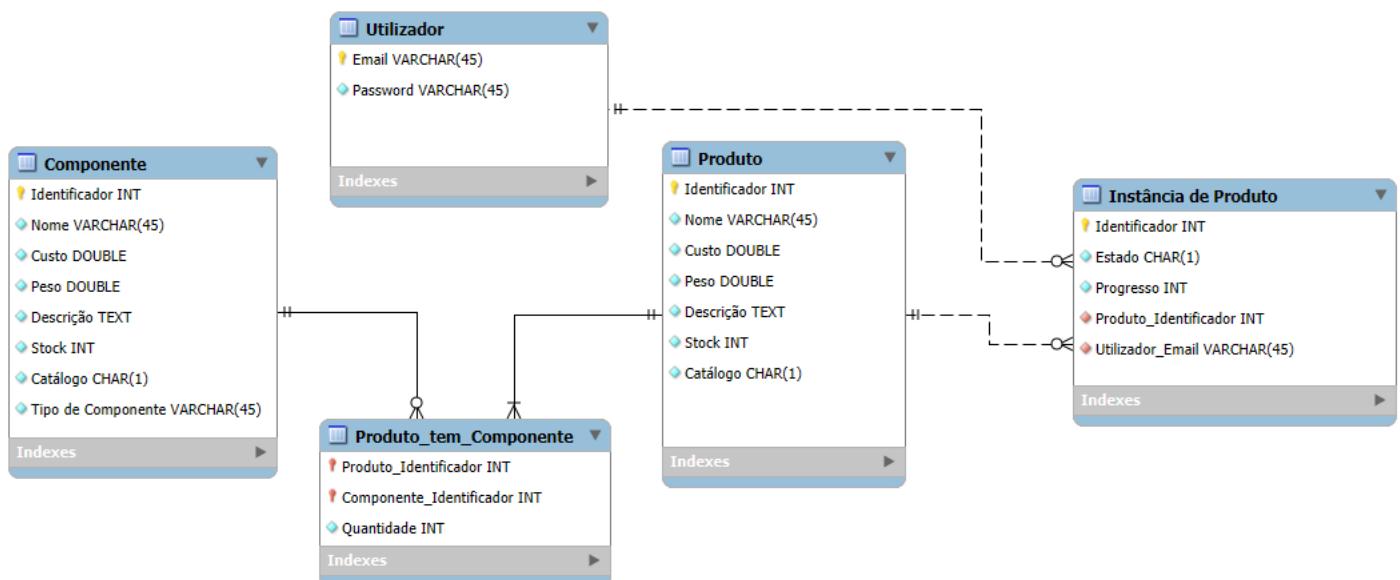


Figura 4.1: Modelo Lógico

4.2 Descrição detalhada dos vários elementos de dados e seus relacionamentos

Nesta secção vamos delinear o porquê de escolhermos o tipo de dados e implementarmos os relacionamentos que implementamos, no nosso modelo lógico.

Para descrever os elementos de dados criamos uma tabela associada a cada uma das **entidades** do modelo lógico e descrevemos os seus "**Atributos**" e "**Tipo de dados**".

Consideramos **irrelevante** incluir restrições do género "NOT NULL" ou "UNIQUE" nas tabelas uma vez que todos os nossos **requisitos (que não são chaves)** são apenas "NOT NULL", ou seja, têm que conter um valor válido obrigatoriamente. **Chaves primárias** e **estrangeiras** são especificadas como "(CP)" e "(CE)" respetivamente.

Entidade	Atributos	Descrição	Tipo de dados
Componente	Identificador(CP)	Identificador único do componente	INT
	Nome	Nome do componente	VARCHAR(45)
	Custo	Custo monetário(euros) do componente	DECIMAL(5,2)
	Peso	Peso(quilogramas) do componente	DECIMAL(6,3)
	Descrição	Pequena descrição do componente	VARCHAR(200)
	Stock	Número de unidades do componente em stock	INT
	Catálogo	Identifica se o componente está incluído no catálogo	CHAR(1)
	Tipo de Componente	Categoría de peça compatível com o componente	VARCHAR(45)

Tabela 4.1: Entidade Componente

Entidade	Atributos	Descrição	Tipo de dados
Produto	Identificador(CP)	Identificador único do produto	INT
	Nome	Nome do produto	VARCHAR(45)
	Custo	Custo monetário(euros) do produto	DECIMAL(5,2)
	Peso	Peso(quilogramas) do produto	DECIMAL(6,3)
	Descrição	Pequena descrição do produto	VARCHAR(200)
	Stock	Número de unidades do produto em stock	INT
	Catálogo	Identifica se o produto está incluído no catálogo	CHAR(1)

Tabela 4.2: Entidade Produto

Para o "**Custo**" e "**Peso**" decidimos, respetivamente, defini-los como "**DECIMAL(5,2)**" e "**DECIMAL(6,3)**". Para o **custo** temos 2 casas decimais para permitir precisão ao **nível do centímo** e 3 casas inteiras que suportam um máximo de **999 euros** para o custo de qualquer produto/componente individual. Para o **peso** consideramos 3 casas decimais , uma vez

que, uma unidade descreve 1 quilograma e queremos precisão ao nível da **grama singular**, sobrando 3 casas para suportar um peso **máximo de 999 quilogramas** por produto/componente individual.

Entidade	Atributos	Descrição	Tipo de dados
Instância de Produto	Identificador(CP)	Identificador único da instância de produto	INT
	Estado	Caracter que tem um estado de produção associado	CHAR(1)
	Progresso	Valor percentual que descreve o progresso da produção	INT
	Produto_Identificador(CE)	Identificador (CP) do Produto associado à Instância de Produto	INT
	Utilizador_Email(CE)	Email (CP) do Utilizador associado à Instância de Produto	VARCHAR(45)

Tabela 4.3: Entidade Instância de Produto

Entidade	Atributos	Descrição	Tipo de dados
Produto _tem_ Componente	Produto_Identificador(CP)(CE)	Identificador(CP) do Produto associado à entidade	INT
	Componente_Identificador(CP)(CE)	Identificador(CP) do Componente associado à entidade	INT
	Quantidade	Número de unidades (componente) associadas ao produto	INT

Tabela 4.4: Entidade Produto_tem_Componente

"**Produto_Identificador**" e "**Componente_Identificador**" representam Chaves Estrangeiras individuais para um "Produto" e "Componente", respectivamente, e não uma Chave Estrangeira composta.

Quanto aos relacionamentos, criamos também tabelas que descrevem cada um deles com as seguintes colunas :

- **Entidades** - Entidades envolvidas no relacionamento
- **Obrigatoriedade** - Indica se a entidade associada tem obrigatoriedade de participar no relacionamento para ser válida , 1 é o valor afirmativo e 0 a negação
- **Cardinalidade** - Indica o número de ocorrências de uma entidade que podem ou devem estar associadas às ocorrências da segunda entidade do relacionamento
- **Atributos** - Atributos próprios do relacionamento
- **Chaves** - Chaves estrangeiras associadas a cada entidade participante do relacionamento
- **Entidade** - Entidades associadas às chaves estrangeiras em "**Chaves**"

Relacionamento					
Entidades	Obrigatoriedade	Cardinalidade	Atributos	Chaves	Entidade
Produto	0	1	—	—	—
Instância de Produto	1	n	—	Produto_Identificador	Chave estrangeira para Produto

Tabela 4.5: Relacionamento Produto-Instância de Produto

Relacionamento					
Entidades	Obrigatoriedade	Cardinalidade	Atributos	Chaves	Entidade
Utilizador	0	1	—	—	—
Instância de Produto	1	n	—	Produto_Identificador	Chave estrangeira para Produto

Tabela 4.6: Relacionamento Utilizador-Instância de Produto

Relacionamento (Produto_tem_Componente)					
Entidades	Obrigatoriedade	Cardinalidade	Atributos	Chaves	Entidade
Produto	1	n	Quantidade - INT	Componente_Identificador	Chave estrangeira para Componente
Componente	0	m		Produto_Identificador	Chave estrangeira para Produto

Tabela 4.7: Relacionamento Produto_tem_Componente

O **relacionamento** entre “**Produto**” e “**Instância de Produto**” permite-nos obter várias informações sobre o tipo de produto que uma certa instância de produto engloba. Podemos minimizar, desta forma, a complexidade de ter vários produtos ao distinguir as instâncias dos mesmos de forma minimalista. Este relacionamento **não é obrigatório** para o “**Produto**” pois podemos ter vários produtos que nunca foram produzidos.

O **relacionamento** entre “**Utilizador**” e “**Instância de Produto**” existe para permitir distinguir o utilizador específico que criou as instâncias, ou seja, permite saber quem iniciou a produção de qualquer produto na linha de montagem. Este relacionamento **não é obrigatório** para o “**Utilizador**” pois podemos ter vários utilizadores que não iniciaram uma produção.

Finalmente o **relacionamento** entre o “**Produto**” e “**Componente**”, é mais complexo que os anteriores com a sua cardinalidade de n para m ou “muitos para muitos”. Este relacionamento é essencial para garantir que um produto esteja associado aos componentes individuais que o compõem, com um registo do número de unidades necessárias de cada um deles para a montagem do mesmo. O relacionamento dispõe de uma cardinalidade n para m para permitir que um mesmo componente componha um ou mais produtos distintos e que podemos registar a “quantidade” de cada um deles como um atributo associado. Este relacionamento **não é obrigatório** para o “**Componente**” pois podemos ter componentes no sistema que não sejam utilizados para qualquer produto.

5 Esboço das Interfaces do Sistema

5.1 Estrutura geral das interfaces do sistema

Com base em toda a informação prévia e com uma ideia agora bem definida sobre o que será desenvolvido, decidimos criar esboços para ilustrar os "layouts" que a equipa pretende implementar. Deste modo podemos apresentar a nossa visão do que será o programa final e garantir que esta se alinha com a visão de Inácio.

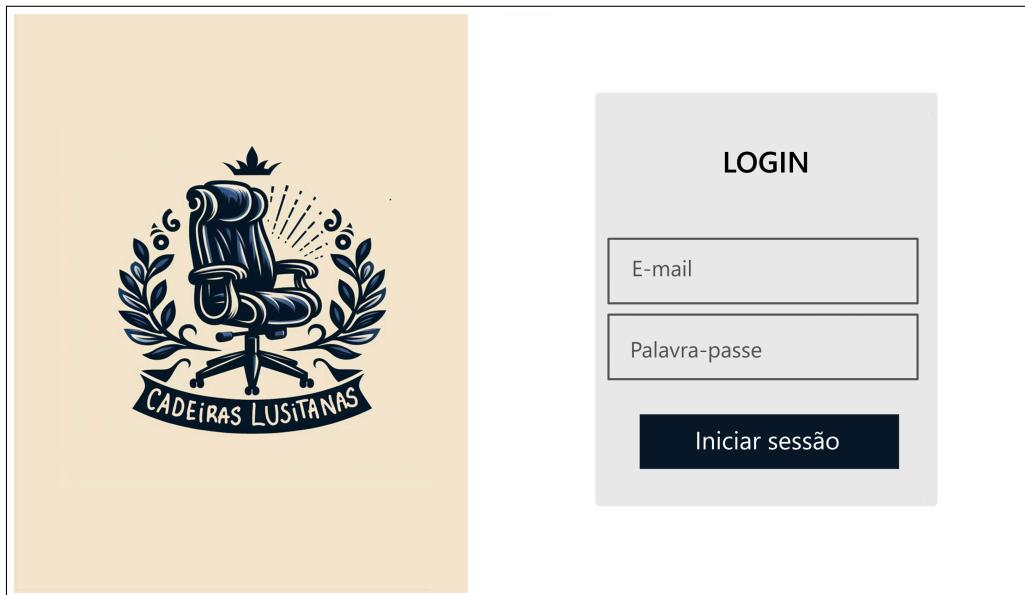


Figura 5.1: Menu "Iniciar sessão"



Página inicial

Gestão de Produtos

Gestão de Componentes

Terminar sessão

Histórico de produção

Identificador	Produto	Peso	Custo	Quantidade
3	Produto C	20kg	120€	2
2	Produto B	22kg	174€	4
3	Produto C	20kg	180€	3

Estatísticas

Produto	Total produzido	Tempo médio de produção	Custo Total de produção
Produto A	3	33s	222€
Produto B	4	37s	174€
Produto C	5	41s	300€

Figura 5.2: Menu "Página inicial"



Página inicial

Gestão de Produtos

- Producir produto
- Visualizar linha de montagem
- Criar novo produto
- Diminuir stock de produto
- Remover produto

Gestão de Componentes

Terminar sessão

Catálogo de Produtos

Identificador	Nome	Peso	Custo	Stock
1	Produto A	20kg	56€	Ver descrição 5
2	Produto B	22kg	65€	Ver descrição 2

Figura 5.3: Menu "Catálogo de Produtos"



Página inicial

Gestão de Produtos

Gestão de Componentes
Criar novo componente
Diminuir stock de componente

Terminar sessão

Identificador	Nome	Peso	Tipo	Custo	Stock	Ação
1	Comp. A Ver descrição	0.5kg	Braço	10€	5	+
2	Comp. B Ver descrição	0.7kg	Pistão	20€	9	+

Figura 5.4: Menu "Catálogo de Componentes"



Página inicial

Gestão de Produtos

Gestão de Componentes
Criar novo componente
Diminuir stock de componente

Terminar sessão

Criar novo Componente

Nome

Descrição

Peso (Kgs)

Seleccione o Tipo

Custo (€)

Adicionar Componente

Figura 5.5: Menu "Criar novo Componente"



Página inicial

Gestão de Produtos

- Producir produto
- Visualizar linha de montagem
- Criar novo produto
- Diminuir stock de produto
- Remover produto

Gestão de Componentes

Terminar sessão

Criar novo Produto

Selecione os Componentes

Figura 5.6: Menu "Criar novo Produto"



Página inicial

Gestão de Produtos

- Producir produto
- Visualizar linha de montagem
- Criar novo produto
- Diminuir stock de produto
- Remover produto

Gestão de Componentes

Terminar sessão

Linha de montagem

Identificador: 1 Produto A (1)	Identificador: 1 Produto A (2)	Identificador: 2 Produto B (1)
Componentes:	Componentes:	Componentes:
Componente A ✓	Componente A ✓	Componente A ✓
Componente B ✓	Componente B ✓	Componente B ✓
Componente C ✓	Componente C ✓	Componente C ✓
Componente D	Componente D	Componente D

Tempo: 00:28 Tempo: 00:19 Tempo: 00:10

Tempo médio: 00:33 Tempo médio: 00:33 Tempo médio: 00:37

Fila de espera

Produto C
quantidade: 5

Figura 5.7: Menu "Visualizar linha de Montagem"



Criar novo Utilizador

E-mail

Palavra-passe

Adicionar Utilizador

Página inicial

Gestão de Produtos

Gestão de Componentes

Criar novo utilizador

Terminar sessão

This screenshot shows the 'Criar novo Utilizador' (Create new User) page. On the left, there's a sidebar with links: 'Página inicial', 'Gestão de Produtos', 'Gestão de Componentes', 'Criar novo utilizador' (which is highlighted in blue), and 'Terminar sessão'. The main area has two input fields: 'E-mail' and 'Palavra-passe'. A large dark button at the bottom right says 'Adicionar Utilizador'. Above the input fields, the title 'Criar novo Utilizador' is displayed.

Figura 5.8: Menu "Criar novo Utilizador"

Insira o Identificador

Identificador

Continuar

This screenshot shows the 'Insira o Identificador' (Insert Identifier) page. It features a single input field labeled 'Identificador' and a dark 'Continuar' (Continue) button below it.

Figura 5.9: Menu "Inserir Identificador"

Insira a quantidade

Quantidade

Continuar

This screenshot shows the 'Insira a quantidade' (Insert Quantity) page. It features a single input field labeled 'Quantidade' and a dark 'Continuar' (Continue) button below it.

Figura 5.10: Menu "Inserir Quantidade"

5.2 Caracterização das interfaces

Menu "Iniciar Sessão": Este esboço surgiu do **Use Case nº 2, Iniciar sessão**, e representa o processo de autenticação dos utilizadores no sistema. Para iniciar sessão, estes devem inserir o seu "**E-mail**" e respetiva **palavra-passe** e, por fim, clicar no botão Iniciar Sessão.

Menu "Página inicial": A **página inicial** apresenta o histórico de produção dos últimos produtos produzidos, assim como as respetivas quantidades produzidas. Adicionalmente, são exibidas estatísticas de todos os produtos, incluindo o **total produzido, tempo médio de produção e custo total de produção**. Este menu ficará disponível aos utilizadores imediatamente após o iniciar da sessão. Está associado ao **Use Case nº10**.

Menu "Catálogo de Produtos": A interface "**Catálogo de Produtos**" é acedida clicando diretamente em "Gestão de Produtos". Aqui aparecerão **todos os produtos já inseridos no sistema**, juntamente com os respetivos **atributos** e com um botão que apresentará a descrição de cada um. Este esboço está associado ao **Use Case nº 14**.

Menu "Catálogo de Componentes": O esboço **Catálogo de Componentes**, proveniente dos **Use Case nº6 e nº9**, pode ser acedido clicando em "**Gestão de Componentes**" e é semelhante ao esboço "**Catálogo de Produtos**", com a **adição** de uma funcionalidade para **adicionar stock** a um componente. Esta operação é realizada clicando no botão "+" no lado direito de cada um dos componentes. Após clicar neste botão aparecerá um "**pop-up**", onde deve ser **inserido um valor** válido correspondente ao **stock** que se **pretende adicionar**.

Menu "Criar novo Componente": Referente ao **Use Case nº4**, surgiu a necessidade de criar um menu para a criação de um **novo componente**. Este esboço pode ser acedido clicando em "**Criar novo componente**" e apresenta campos onde devem ser **inseridas** corretamente todas as **informações solicitadas** para realizar esta operação. Por fim, basta clicar no botão "**Adicionar Componente**" para que este seja registado no sistema.

Menu "Criar novo Produto": Do mesmo modo que existe a necessidade de inserir novos componentes é também necessário inserir novos produtos. O menu "**Criar novo Produto**" referente ao **Use Case nº5** é muito semelhante ao menu "**Criar novo Componente**", apresenta campos onde devem ser **inseridas corretamente** informações necessárias para especificar o produto. Para finalizar basta clicar no botão "**Adicionar Produto**" para que este seja registrado no sistema. Este menu é acedido clicando em "**Criar novo produto**" "**sub-menu**" de "**Gestão de Produtos**".

Menu "Remover produto": O esboço "**Remover Produto**" surge da necessidade de remover produtos do catálogo (**Use Case nº11**). Nos "**sub-menus**" de "**Gestão de Produtos**" existe um botão denominado "**Remover produto**". Este apresenta um novo menu, com uma caixa onde o utilizador deve inserir um **identificador** (**Menu "Inserir Identificador"**). Caso este seja **válido**, o sistema irá **remover** o **produto** associado do catálogo de produtos. Após o sucesso ou falha da operação, o sistema irá **redirecionar** o utilizador para o **menu "Gestão de Produtos"**.

Menu "Diminuir stock": Os menus "**Diminuir stock de produto**" e "**Diminuir stock de componentes**" surgem dos **Use Case nº12 e nº13**, respetivamente. Nestes menus, o

utilizador deve inserir o **identificador do componente** (**Menu "Inserir Identificador**) ou do **produto** que deseja diminuir. Caso este seja válido, surge um **novo menu** para inserir a **quantidade** que o utilizador pretende remover (**Menu "Inserir Quantidade**). Por fim, clicando no botão "**Continuar**", o utilizador será redirecionado para o menu "**Gestão de Produtos**" ou "**Gestão de Componentes**", conforme o tipo de processo escolhido (produto ou componente).

Menu "Produzir Produto": O menu "**Produzir Produto**" surge do **Use Case 7**. Neste menu o utilizador deve inserir o **identificador do componente** (**Menu "Inserir Identificador**) do **produto** que deseja produzir. Em seguida abre-se um novo menu no qual o utilizador deve especificar a quantidade de produto a produzir (**Menu "Inserir Quantidade**”).

Menu "Linha Montagem": O menu "**Linha Montagem**" surge do **Use Case 8**. Para aceder a este menu basta clicar na opção "**Visualizar Linha de Montagem**" que deverá aparecer na esquerda da tela.

Menu "Criar novo Utilizador": O menu "**Criar novo Utilizador**" surge do **Use Case 1** e apenas se encontra disponível para o **Administrador**. Para aceder a este menu basta clicar na opção "**Criar novo Utilizador**" que deverá aparecer na esquerda da tela. Após isso o utilizador terá de introduzir o **E-mail e Palavra-Passe** do novo utilizador e em seguida pressionar o botão **Adicionar Utilizador**.

6 Implementação da Aplicação

6.1 Apresentação e descrição do processo de implementação realizado.

Antes de seguirmos com a implementação do sistema, decidimos que seria importante definir o modelo arquitetónico a ser adotado. Esta decisão surge com o intuito de obter uma base sólida para o desenvolvimento do sistema, garantindo que todos os seus componentes estejam organizados de forma coerente. Deste modo, garantimos que o sistema fica estruturado de forma a suportar futuras necessidades de expansão.

Após uma análise cuidadosa, optamos por seguir uma arquitetura em três camadas ("**three layer architecture**").

Com esta escolha, temos uma separação das diferentes lógicas do sistema - **Interface, negócio e dados** - promovendo a **modularidade** do nosso projeto e a já mencionada **escalabilidade**. Para além disso com a utilização desta arquitetura conseguimos facilmente **dividir responsabilidades** por diferentes equipas dentro da empresa, permitindo a criação de grupos especializados. A Figura 6.1 ilustra a nossa planificação inicial para a organização do sistema.

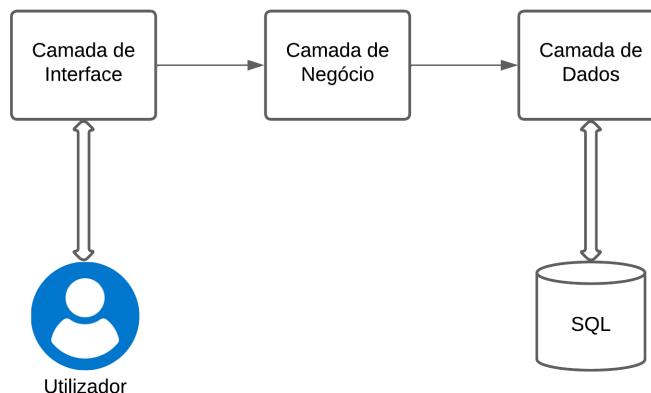


Figura 6.1: Arquitetura Inicial

Com a arquitetura definida demos inicio à implementação do sistema. Para tal começamos pela criação da base de dados "**blazorDB**".

De seguida criamos o projeto e as pastas que irão conter a lógica das diferentes camadas, tal como representado na figura 6.2.

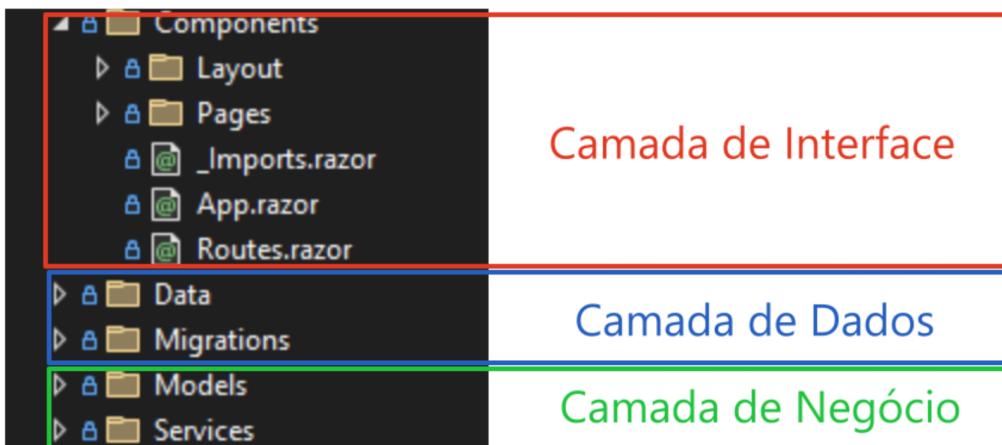


Figura 6.2: Pastas do projeto

No que toca ao método de implementação, decidimos adotar uma implementação sequencial, que começou com o desenvolvimento dos utilizadores. Estes foram os primeiros a ser implementados pois sabíamos que seria uma das tarefas mais simples. Inicialmente criamos a respetiva tabela na base de dados e implementamos as funcionalidades descritas nos USE CASE 2, 3 e 1, **Iniciar Sessão, Terminar Sessão e Criar Utilizador**. Para melhorar o controlo administrativo decidimos incluir funcionalidades não descritas nos USE CASE, nomeadamente, **Editar Utilizador e Remover Utilizador**. Consideramos estas duas novas funcionalidades essenciais para que os administradores consigam corrigir eventuais erros que possam surgir sobre utilizadores.

O passo seguinte foi a implementação dos componentes. Estes foram escolhidos devido à dependência lógica que os produtos têm deles (componentes compõe produto). Mais uma vez, começamos por criar a respetiva tabela na base de dados. Implementamos as funcionalidades descritas nos USE CASE 4,6,9 e 13, **Criar novo Componente, Abastecer Stock, Visualizar catálogo de componentes e Diminuir stock de Componentes**, respetivamente, com a introdução adicional do parâmetro **tempo** no componente, parâmetro que estava em falta na descrição do USE CASE 4. Sem este parâmetro seria impossível calcular o tempo de montagem de um produto. Para além disso decidimos mais uma vez adicionar as opções de editar e remover componente de modo a que seja possível refletir eventuais ajustes de preços e correções de stock.

Ao longo do desenvolvimento dos componentes surgiu um problema que considerámos importante resolver. Todos os integrantes do grupo tinham que fazer mudanças no caminho para a base de dados de modo a que o sistema pudesse executar corretamente na sua máquina. Para resolver este problema chegámos à conclusão que a utilização de "container's" via **Docker** seria apropriado (Manual de "Set-up" da Aplicação com Docker em anexo).

Finalmente passamos à implementação dos produtos. Criamos a tabela correspondente na base de dados e implementamos as funcionalidades básicas dos produtos, sendo estas **Visualizar Catálogo de Produtos**, **Criar novo Produto**, **Remover Produto do Catálogo de Produtos**, **Diminuir stock de Produto** referentes aos USE CASE 14,5,11 e 12 respetivamente.

Feito isto passamos finalmente à lógica principal do sistema, a **Linha de Montagem**. Com o objetivo de otimizar o processamento do produto decidimos estruturar a linha de montagem em três fases sequenciais.

Na **1^a fase** que representa **30%** do processo de montagem, as rodas da cadeira são unidas com o pistão. Assim que esta fase termine o produto 1 avança para a **2^a fase** e em **simultâneo** o produto 2 entra na **1^a fase**, permitindo uma estrutura de **pipeline** que reflete o processo de montagem real. A **2^a fase** representa **50%** do trabalho de montagem, visto que é responsável por unir o encosto à almofada e ao conjunto montado na **1^a fase**. Finalmente, a **3^a fase** do nosso processo de montagem representa os últimos **20%** de progresso. Nesta fase os braços são adicionados ao produto completando-o.

Para além desta linha de montagem, possuímos uma fila de espera que funciona por ordem **FIFO**(First-in-First-Out). Assim que a **1^a fase** se encontre vazia, o primeiro produto na fila de espera é processado. É importante notar que os stocks dos componentes são **automaticamente removidos** antes da produção estar completa, representando a "**reserva**" dos mesmos para produção enquanto que os produtos só serão registados na contagem de stock ao concluir o seu processo de produção.

O processo de montagem descrito está representado na imagem abaixo :

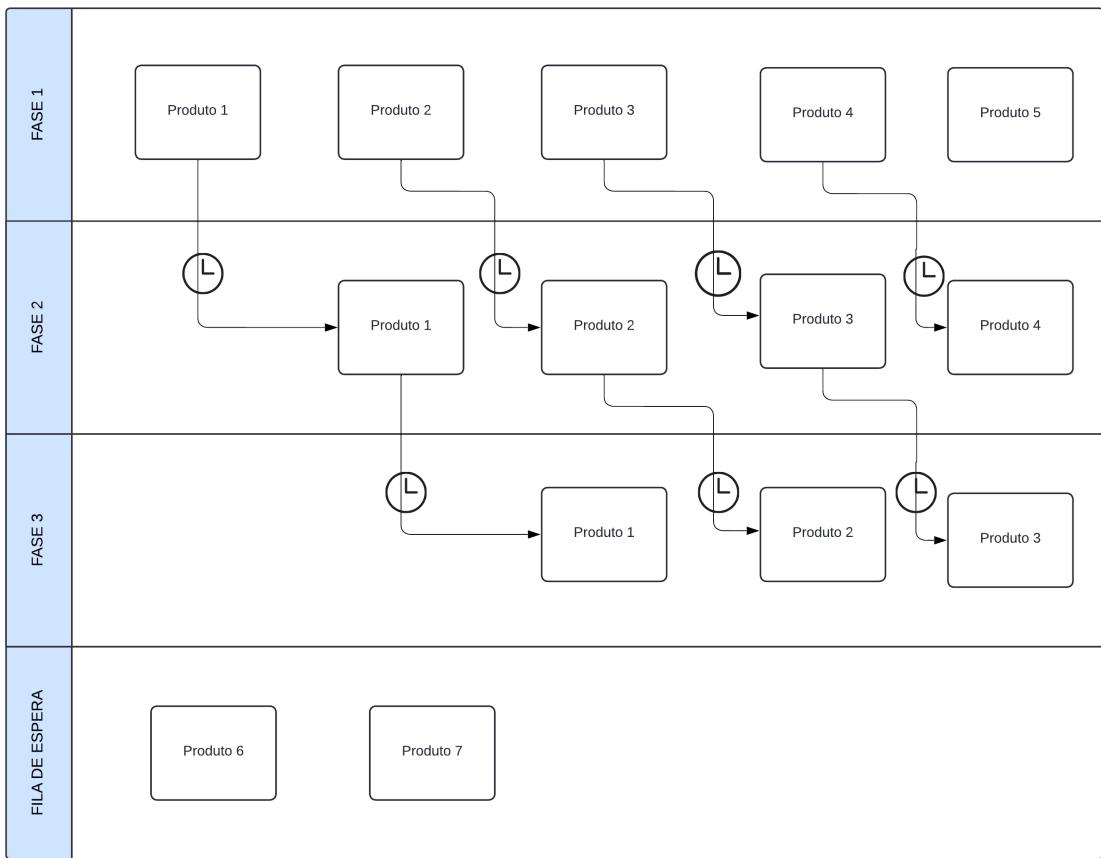


Figura 6.3: Funcionamento da linha de montagem

Tendo o modelo em mente passamos à implementação da linha de montagem no sistema, referente ao USE CASE 8. Para assegurar a robustez do sistema, utilizamos o conceito de **instâncias de produto**, permitindo gerir múltiplas cópias de um mesmo produto com diferentes estados e progressos. As instâncias são armazenadas na base de dados, garantindo a **persistência** do estado da produção mesmo em caso de **falha** do sistema, além de servirem como **histórico de produção**.

Com a linha de montagem já pronta, iniciámos a implementação do USE CASE 7, **Producir Produto** e modificámos a página inicial para que agora represente o USE CASE 10, **Visualizar histórico de produção**.

Relativamente à **base de dados** inicialmente conceptualizada (Figura 4.1), a nossa solução final apresenta várias divergências em relação ao planeado. A grande maioria destas mudanças tratam-se de pequenos extras ou modificações de valores que servem um propósito idêntico àquele planeado, sendo as principais mudanças:

- Foi adicionada uma variável "**role**" ao utilizador para que seja possível identificar o tipo de utilizador. Inicialmente planeámos distinguir utilizadores pelo seu endereço de email mas desta forma facilitamos esse processo.

- O atributo "Quantidade" na tabela "**Produto_tem_Componente**" foi movido para a tabela "Produto" e substituído pelo atributo "**n_rodas**". Decidimos fazer esta modificação pois a única peça que pode ser repetida em uma cadeira é a roda.
- Adicionamos um novo atributo "**Tempo**" na instância de produto e no componente para definir um tempo "aleatório" de montagem alusivo a cenários reais.
- Alguns atributos foram alterados, nomeadamente os valores de tipo "Decimal" passaram a "**real**". Para além disso, o atributo "**Catálogo**", tanto da entidade "Componente" como "Produto", passou a denominar-se "**Visível**" e trata-se de um valor "**int**".
- Todos os atributos do tipo "**varchar**" passaram a ser "**nvarchar**" permitindo maior diversidade de caracteres especiais.

Com a aplicação já desenvolvida, o nosso modelo inicial (Figura 6.1) tem de ser atualizado, de modo a refletir tudo o que foi implementado. A Figura 6.3 demonstra como o projeto ficou organizado pós-implementação.

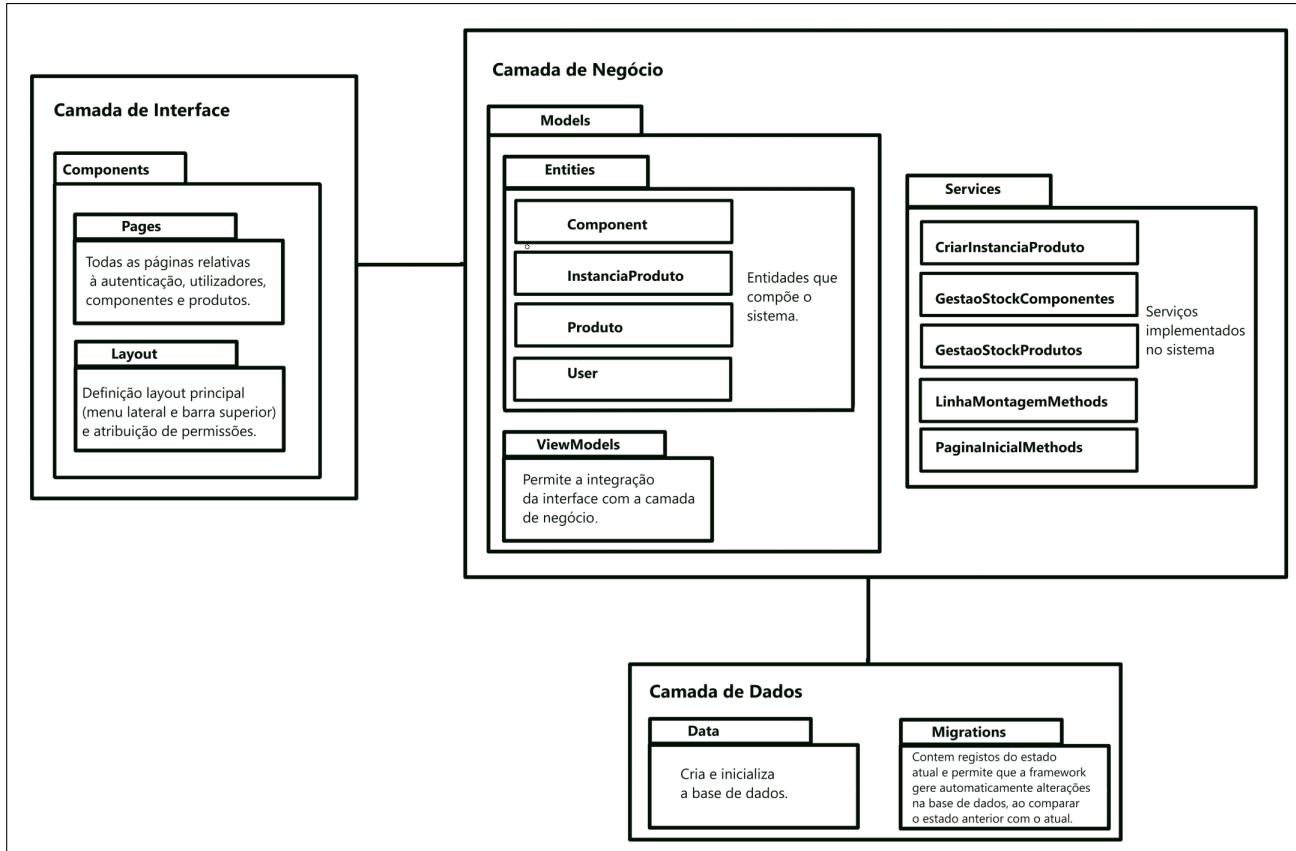


Figura 6.4: Arquitetura final

6.2 Apresentação da aplicação e explicação dos serviços implementados.

Ao longo do processo de implementação tivemos o devido cuidado em respeitar os "mockups" definidos previamente, implementando interfaces paralelas a estes em termos estéticos e funcionais, com a exceção de algumas funções adicionais que sustentam uma experiência de uso mais intuitivo no sistema. Com isto segue-se uma exposição descritiva da aplicação.

Ao iniciar a aplicação são requisitadas as credenciais do utilizador numa página de "**login**" referente ao "mockup - **Iniciar Sessão**" (Figura 5.1) (Use Case 2). Só ao iniciar uma sessão com sucesso é que o utilizador ganha acesso a qualquer outra funcionalidade.

Dada como concluída a autenticação, o utilizador é redirecionado para a página inicial, na qual se podem visualizar o "**Histórico de produção**" e as "**Estatísticas**" referentes ao "mockup" "**Página Inicial**"(Figura 5.2) (Use Case 10). Caso ainda não tenha sido criado nenhum produto, a secção de estatísticas será substituída pela mensagem "**Ainda não existem produtos**", e de forma semelhante, caso não exista nenhum produto em produção ou já produzido a secção do Histórico irá ser substituída pela mensagem "**Ainda não foram produzidos produtos**". Para além disso, o menu lateral destacado na **Figura 6.4** permite o acesso fácil às diversas funcionalidades implementadas. É importante destacar que a opção "**Terminar Sessão**" (Use Case 3) pode ser utilizada em qualquer página (pós inicio de sessão) e redireciona o utilizador para a página de iniciar sessão.

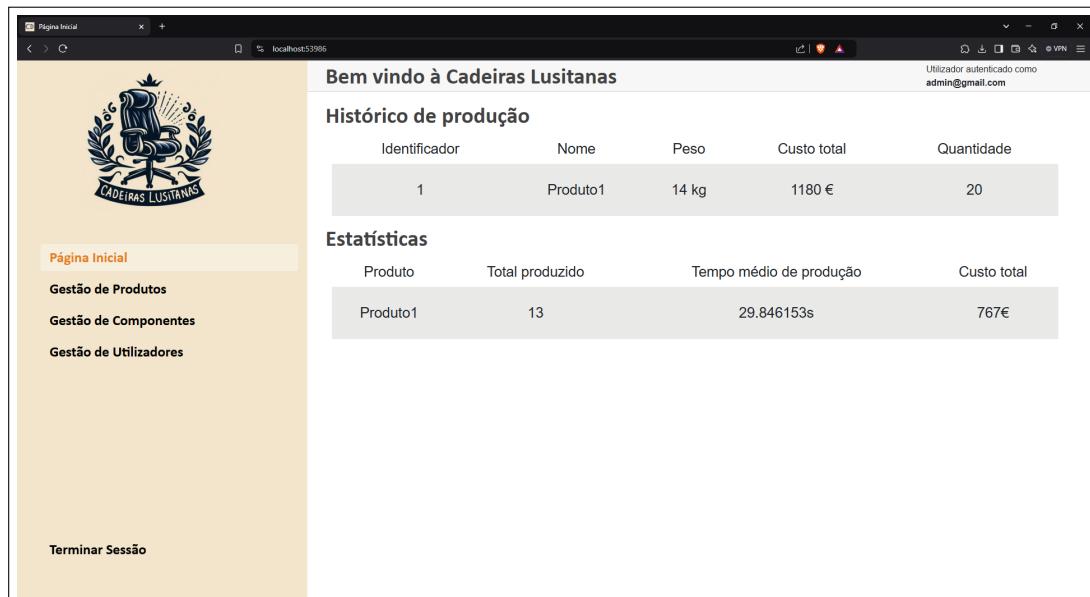


Figura 6.5: Página Inicial

Ao selecionar a opção "**Gestão de Produtos**" no menu lateral é apresentado o "**Catálogo de Produtos**", referente ao "mockup" com o mesmo nome (Figura 5.3) (Use Case 14), e também, no menu lateral, 4 opções de serviços associados a produtos:

- Produzir produto
- Visualizar linha de montagem
- Criar novo produto
- Diminuir stock de produto

Quando o catálogo se encontra vazio é apresentada uma mensagem a destacar esse facto, senão é apresentada a lista de produtos (Figura 6.5) , alguns detalhes relevantes, um botão "**C**" que permite visualizar os componentes que compõe o produto e uma opção "**Remover**" que pode eliminar o produto do catálogo após uma página de confirmação (Use Case 11).

The screenshot shows a web browser window titled 'Catálogo de Produtos' at 'localhost:53986/products'. The main content area displays a heading 'Bem vindo à Cadeiras Lusitanas' and 'Catálogo de Produtos'. Below this is a table with one row, showing a single product entry:

Identificador	Nome	Componentes	Peso	Custo	Stock	
1	Produto1 Ver descrição	C	14kg	59€	4	Remover

The left sidebar contains a logo for 'Cadeiras Lusitanas' and a navigation menu with the following items:

- Página Inicial
- Gestão de Produtos** (highlighted)
- Produzir produto
- Visualizar linha de montagem
- Criar novo produto
- Diminuir stock de produto
- Gestão de Componentes**
- Gestão de Utilizadores**
- Terminar Sessão

Figura 6.6: Catálogo de Produtos

Ao selecionar "**Produzir produto**", que se trata de uma junção entre dois mockups, "**Inserir Identificador**" e "**Inserir Quantidade**" (Figuras 5.9 e 5.10, respetivamente) (Use Case 7), o programa permite que, através de um identificador válido, se dê inicio ao processo de produção de um produto; consequentemente o utilizador é redirecionado para a "**Linha de montagem**" onde o pedido será refletido, sendo esta página proveniente do "mockup" "**Visualizar Linha de Montagem**" (Figura 5.7) (Use Case 8). Na linha de montagem (Figura 6.6) pode-se verificar a "Fila de espera" com todos os produtos que ainda não iniciaram o processo de montagem e a "Linha de Montagem" com 3 secções que permitem produção em "**pipeline**" de até 3 produtos. Cada uma das secções apresenta o Identificador da instância do produto, o nome e identificador do produto, os componentes instalados e por instalar, o tempo decorrido, a média de tempo de montagem total e uma barra de progresso alusiva aos mesmos.

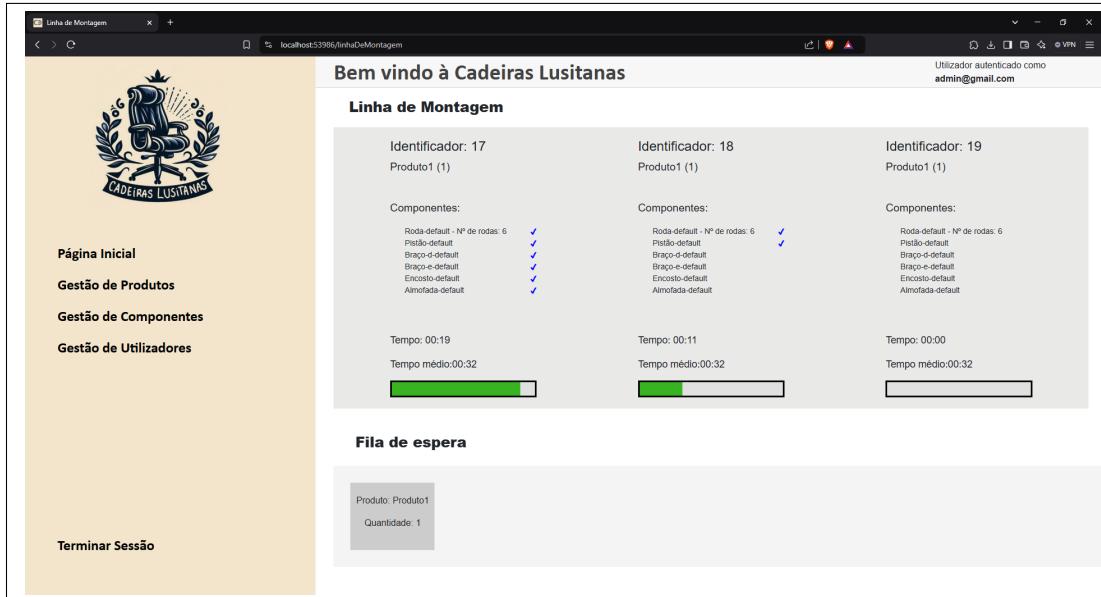


Figura 6.7: Linha de Montagem

É possível criar e diminuir o stock de um produto, que referem respetivamente ao mockup "**Criar Novo Produto**" (Figura 5.6) (Use Case 5) e à semelhança do "**Producir produto**"(Figuras 5.9 e 5.10) (Use Case 7). Ao criar são requisitados todos os campos de um produto exceto peso e custo, pelo que estes são calculados através dos componentes selecionados, tal como o tempo de montagem. Para diminuir o stock de um produto basta identificar um produto e valor válido e confirmar o pedido.

Na secção de "**Gestão de Componentes**", baseado no mockup "**Catálogo de Componentes**" (Figura 5.4) (Use Case 9), é mais uma vez apresentado o catálogo associado como página principal (Figura 6.7). Aqui temos uma lista dos componentes com informações relevantes e, para além de permitir remover componentes do catálogo, a tabela apresentada permite **aumentar o stock** (Use Case 6) e **editar** detalhes dos mesmos.

Ao **editar** os componentes podemos substituir qualquer parâmetro por um valor válido, com exceção do identificador, que é permanente após a criação inicial, e o stock que apenas pode ser alterado ao utilizar as funcionalidades de aumento e diminuição explicitamente. Para além disso, a edição de um componente, irá de forma **automática atualizar** os produtos a que o componente está associado. Tanto a função de remoção como de edição de componentes não foram planeadas mas, dada a fácil implementação e utilidade, decidimos inclui-las.

The screenshot shows a web application titled "Catálogo de Componentes" (Catalogue of Components) for "Cadeiras Lusitanas". The interface includes a sidebar with links for "Página Inicial", "Gestão de Produtos" (selected), and "Gestão de Utilizadores". The main content area displays a table of components:

Identificador	Nome	Peso	Tipo	Custo	Stock	Tempo (seg)	Editar	+	Remover
1	Roda-default <small>Ver descrição</small>	1kg	RODA	5€	40	1-2	Editar	+	Remover
2	Pistão-default <small>Ver descrição</small>	5kg	PISTÃO	15€	90	3-5	Editar	+	Remover
3	Braço-d-default <small>Ver descrição</small>	2kg	BRAÇO_DIREITO	10€	90	2-3	Editar	+	Remover
4	Braço-e-default <small>Ver descrição</small>	2kg	BRAÇO_ESQUERDO	10€	90	2-3	Editar	+	Remover
5	Encosto-default <small>Ver descrição</small>	3kg	ENCOSTO	15€	90	4-5	Editar	+	Remover
6	Almofada-default <small>Ver descrição</small>	1kg	ALMOFADA	4€	90	1-2	Editar	+	Remover
7	Almofada-default2 <small>Ver descrição</small>	1kg	ALMOFADA	4€	100	1-2	Editar	+	Remover

Figura 6.8: Catálogo de Componentes

Nos dois serviços adicionais disponibilizados na "**Gestão de Componentes**" é possível criar um novo componente, referente ao mockup "**Criar Novo Componente**" (Figura 5.5)(Use Case 4) e **diminuir o stock**, referente à junção de dois mockups (Figuras 5.9 e 5.10)(Use Case 13). A criação de componentes e diminuição de stock estão associados a várias verificações e, no caso da manipulação de stock, há uma confirmação explícita que dá ênfase ao impacto da ação.

Finalmente os serviços associados à "**Gestão de Utilizadores**", exclusivos a administradores (Utilizadores do tipo "ADMIN"), permitem o acesso e uso de todas as suas funcionalidades através da página principal "/users"; nesta página pode-se **visualizar** as informações associadas a todas as contas registadas, **editar** essas informações, **remover** as contas do sistema e **criar** novos utilizadores (Use Case 1). Mais uma vez implementamos novas funcionalidades que tornam o uso do sistema mais prático, neste caso a possibilidade de editar e remover contas já registadas.

6.3 Análise e avaliação da aplicação desenvolvida.

A aplicação foi desenvolvida com o intuito de corresponder a todos os requisitos e USE CASE levantados e, após o desenvolvimento, podemos confirmar que suporta todas as funcionalidades planeadas:

Administradores:

- Criar utilizador;

Utilizadores e Administradores:

- **Componentes:**

- Criar componentes;
 - Gerir stocks de componentes;
 - Visualizar catálogo de componentes.

- **Produtos:**

- Criar produtos;
 - Produzir produtos;
 - Gerir catálogo de Produtos;
 - Visualizar linha de montagem;
 - Consultar histórico e estatísticas sobre produtos;
 - Diminuir stock de produtos.

No caso dos administradores, para além da criação de utilizador, de acordo com aquilo que foi planeado, foram adicionadas três novas funcionalidades: **Remover Utilizador**, **Editar Utilizador** e **Visualizar Utilizador** que permitem uma manipulação direta dos dados associados a utilizadores.

No que toca às funcionalidades relativas aos componentes, todas foram implementadas, no entanto, a criação de componente sofreu uma pequena alteração. Agora necessita receber um **intervalo de tempo** de produção, de modo a possibilitar o cálculo do tempo total de produção de um produto. Para além disso implementamos funcionalidades extra: **Remover Componente** e **Editar Componente**.

Finalmente, a respeito das funcionalidades relativas aos produtos, é muito importante ressaltar que a opção de **visualizar o histórico e estatísticas** não funciona do modo inicialmente planeado. Neste momento, como estas são calculadas sempre que a página inicial é **recarregada**, a alteração de um componente afeta produtos já produzidos, o que consequentemente **afeta** o

histórico e estatísticas sobre os mesmos. Este comportamento difere do planeado pelo que apenas os produtos produzidos após a edição deviam sofrer com as devidas alterações.

Limitações:

Apesar de atendermos ao requisitos levantados, a aplicação apresenta algumas limitações relevantes.

Não houve um foco em **segurança**, tendo apenas sido implementadas funcionalidades básicas, como a restrição de acesso a funcionalidades através de acesso direto por URL.

A funcionalidade de visualizar linha de montagem **não possui** um sistema de **recarregamento automático**, pelo que atualmente, para que a linha de montagem reflita o estado atual das montagens, é necessário que o utilizador recarregue manualmente a página.

Como mencionado, o **histórico e estatísticas** são recalculados com base nos dados dos produtos no momento atual, pelo que, após edições em componentes, os valores de custo total e peso não vão refletir os valores do momento de produção.

Ainda que tenhamos integrado containers **Docker** e **Cookies** na aplicação, não conseguimos resolver a **colisão** entre os dois. Consequentemente, a utilização de Cookies para guardar os dados do utilizador registado apenas funciona quando o Docker é desativado.

A aplicação não foi desenvolvida com dispositivos de **diferentes proporções** em mente, como tal, a utilização da aplicação em dispositivos a usufruir de ecrãs demasiado pequenos ou de proporções significativamente diferentes de uma proporção 16:9 **não é suportada**.

Finalmente, utilizamos a "Entity Framework" como forma de acesso à base de dados o que **não nos permite a utilização direta de comandos "SQL"**. Utilizar uma Framework que permita esta utilização pode ser vantajoso em termos de performance e personalização de acesso à base de dados.

7 Conclusão

A aplicação desenvolvida conseguiu atender aos requisitos e USE CASE definidos no início do projeto com pouca alteração, sendo que nenhuma das funcionalidades ficou em falta. No entanto, durante o desenvolvimento, surgiram algumas limitações e desafios que poderiam ter sido resolvidos de forma diferente.

Na **primeira fase** do projeto super-estimamos a qualidade do trabalho desenvolvido. Abdicamos de muito tempo para aperfeiçoar detalhes e explicações de conceitos que, no fundo, não representam um valor substancial na documentação produzida. Do mesmo modo deixamos de lado, inconscientemente, um aprofundar necessário em várias vertentes do processo, por exemplo, na conceção da base de dados. A primeira fase foi demorada mas esse tempo não foi aproveitado eficientemente e a atenção requerida passo a passo no processo também não foi assegurada. Apesar das falhas, a base para o desenvolvimento da aplicação foi o suficiente para que, com algumas alterações, pudéssemos alcançar um sistema funcional que, durante o processo de desenvolvimento, nos fez aperceber dos lapsos no planeamento realizado na primeira fase.

Na **segunda fase**, o ponto que considerámos mais importante, é o facto de não termos atribuído à **linha de montagem** a devida importância, tendo esta sido deixada para o fim durante o processo de implementação. Provavelmente, se tivéssemos focado os nossos esforços nesta desde o início, teríamos conseguido implementar uma atualização automática da página que ficou em falta no sistema e também o devido funcionamento do histórico e estatísticas. Para além disso, o nosso esforço inicial ter sido direcionado aos **utilizadores** foi uma péssima estratégia, visto que as funcionalidades associadas apresentam pouca importância para o funcionamento global do sistema. Em relação à **segurança**, ainda que existam restrições básicas de acesso por URL, ficaram em falta mecanismos de proteção mais robustos que salvaguardem a aplicação contra outras vulnerabilidades.

Para uma **fase futura** do trabalho, seria vital dirigir os nossos esforços à correção do funcionamento do histórico e estatísticas e também à implementação da linha de montagem com atualização em tempo real. Seria também crucial a implementação dos já mencionados mecanismos de segurança mais robustos, bem como a habilitação de Cookies com Docker. Finalmente, algo que, embora menos importante poderia também ser implementado, seria a utilização de uma nova "framework" que permita utilização direta de comandos SQL, bem como a garantia do funcionamento da aplicação, independentemente do tipo de ecrã/dispositivo utilizado.

Em suma, ainda que a aplicação esteja funcional e atenda a todos os requisitos principais, existem áreas que necessitam de melhoria e que deverão ser abordadas numa próxima fase do projeto.

8 Referências

- Sommerville, I. (2015). *Software Engineering, Global Edition*.
Microsoft. (n.d.). .NET API browser. Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/api/>

9 Sinónimos

"Software" - Sistema - Aplicação

Produto - Cadeira

Gestor - Administrador

Palavra-passe - Senha