

UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

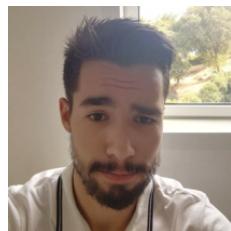
Trabalho Prático
Desenvolvimento de Sistemas de Software

Catarina Machado (a81047) Gonçalo Faria (a86264)
João Vilaça (a82339) José Fernandes (a82467)

6 de Janeiro de 2019



(a) Catarina.



(b) Gonçalo.



(c) João.



(d) José.

Conteúdo

1	Introdução	2
2	Descrição do Enunciado Proposto	2
3	Descrição dos Objetivos do Relatório	3
4	Modelo de Domínio	4
5	Modelo de Use Cases	9
6	Especificação de Use Cases	14
6.1	Configuração Ótima	14
6.2	Criar Encomenda	15
6.3	Iniciar Configuração	16
6.4	Guardar Configuração	16
6.5	Abrir Configuração Guardada	17
6.6	Adicionar Componente	18
6.7	Remover Componente	19
6.8	Adicionar Pacote	20
6.9	Consultar próxima encomenda com material	21
6.10	Consultar informação do componente	22
6.11	Atualizar Stock	23
6.12	Apagar Pacotes	24
6.13	Adicionar Novo Pacote	25
6.14	Adicionar Funcionário	26
6.15	Autenticar	26
7	Idealização do Modelo da Interface com o Utilizador da 1.^a Fase do Projeto	28
7.1	Máquina de estado	28
7.2	Prototipagem da UI	32
8	Máquina de Estado da Configuração	46
9	Diagrama de Sequência de Sistemas	47
9.1	Configuração Ótima	47
9.2	Criar Encomenda	47
9.3	Iniciar Configuração	48
9.4	Guardar Configuração	48
9.5	Abrir Configuração Guardada	48
9.6	Adicionar Componente	49
9.7	Remover Componente	50
9.8	Adicionar Pacote	51
9.9	Consultar próxima encomenda com material	52
9.10	Consultar informação do componente	52
9.11	Atualizar Stock	53
9.12	Apagar Pacotes	53
9.13	Adicionar Novo Pacote	54
9.14	Adicionar Funcionário	54
9.15	Autenticar	55
10	Diagrama de Classes	56
11	Diagrama de ORM	57

12 Diagrama de Packages	58
13 Diagrama de Sequência de Subsistemas	59
13.1 Configuração Ótima	59
13.2 Criar Encomenda	60
13.3 Iniciar Configuração	60
13.4 Guardar Configuração	60
13.5 Abrir Configuração Guardada	61
13.6 Adicionar Componente	61
13.7 Remover Componente	62
13.8 Adicionar Pacote	63
13.9 Consultar próxima encomenda com material	64
13.10 Consultar informação do componente	64
13.11 Atualizar Stock	65
13.12 Apagar Pacotes	65
13.13 Adicionar Novo Pacote	66
13.14 Adicionar Funcionário	66
13.15 Autenticar	67
14 Diagrama de Sequência de Implementação	68
14.1 Configuração Ótima	68
14.2 Criar Encomenda	68
14.3 Iniciar Configuração	69
14.4 Guardar Configuração	69
14.5 Abrir Configuração Guardada	69
14.6 Adicionar Componente	70
14.7 Remover Componente	70
14.8 Adicionar Pacote	71
14.9 Consultar próxima encomenda com material	72
14.10 Atualizar Stock	72
14.11 Apagar Pacotes	73
14.12 Adicionar Novo Pacote	73
14.13 Adicionar Funcionário	73
14.14 Autenticar	74
15 Base de Dados	75
16 Modelo de Programação Inteira para a Configuração Ótima	76
17 Implementação	77
17.1 Implementação Configuração Ótima	77
18 Manual de Utilização	78
19 Análise Crítica do Resultado Obtido da 1.^a Fase do Projeto	96
20 Conclusão Final	97

1 Introdução

Este relatório foi desenvolvido no âmbito da realização do trabalho prático da unidade curricular de Desenvolvimento de Sistemas de Software. O projeto “Car Configurator Hub” consistiu na modelação e implementação de um configurador de carros que permite a criação de configurações, constituídas por componentes individuais ou por pacotes de componentes. Estas configurações darão mais tarde origem a encomendas que serão consultadas pelos funcionários fabris. A aplicação permitirá ainda a gestão de stocks de componentes, pacotes e de utilizadores, entre outras. Para a modelação utilizamos a UML, Linguagem de Modelagem Unificada, para elaborar a grande maioria dos diagramas que compõem este projeto. Para a especificação dos Use Cases foi utilizado o Microsoft Excel e para o modelo lógico da base de dados foi utilizado o MySQL Workbench. Quanto a tecnologias para a aplicação, este configurador será desenvolvido em Java e será executado no desktop, na JVM, no stand pelo funcionário junto ao cliente. A interface gráfica foi desenvolvida recorrendo ao uso do software JavaFX e, por fim, para a base de dados optamos pela utilização da ferramenta MySQL.

2 Descrição do Enunciado Proposto

O enunciado propõe a modelação de um configurador de carros. Este configurador será executado no desktop, na máquina virtual de java, no stand pelo funcionário junto ao cliente.

No processo de configuração serão identificados os componentes básicos. Após esta seleção o cliente terá a possibilidade de escolher componentes opcionais. Os componentes poderão ser incompatíveis ou requerer a inclusão de outros componentes. Quando tal ocorre o sistema deverá avisar que o componente requer ou é incompatível com outro componente sendo que é o cliente que decide se o sistema deve inserir o componente apesar das incompatibilidades que isto pode causar.

O cliente poderá também, com a ajuda do funcionário, selecionar entre vários pacotes. A seleção de um pacote implica a inserção dos componentes que este agrupa na configuração. Certos pacotes poderão oferecer um desconto que é subtraído ao preço final da encomenda e é calculado tendo em conta os componentes agregados. Tal como na seleção dos componentes opcionais podem existir incompatibilidades na adição dos componentes do pacote à configuração, quando isto acontece cabe ao cliente a decisão de prosseguir com a seleção do pacote.

O sistema deverá ainda detetar, após cada adição de componentes, as situações em que o utilizador seleciona todos os componentes pertencentes a um pacote. Quando isto acontece o sistema deverá notificar o utilizador sobre a aquisição do tal pacote.

Em qualquer passo deste processo o cliente poderá pedir ao funcionário para cancelar a configuração, caso o cliente prossiga com o processo de configuração até ao fim ele poderá criar uma encomenda, caso a configuração seja válida. Se o cliente decidir criar a encomenda esta será mandada para uma fila de produção da fábrica.

O sistema oferecerá também uma outra opção de configuração, a de configuração ótima. Para o cliente utilizar esta funcionalidade ele terá primeiro de selecionar os componentes básicos, de seguida o cliente deverá indicar qual a quantia que se encontra disposta a gastar. O sistema deverá então tentar maximizar a utilização da quantia indicada.

Na fábrica os funcionários terão a possibilidade de aceder ao sistema. Caberá aos funcionários da fábrica a tarefa de atualizar os valores dos stocks dos vários componentes. Para que isto seja possível, o sistema terá de guardar para cada componente qual o stock existente. Sempre que os stocks são atualizados o sistema deverá determinar quais as próximas encomendas que podem ser produzidas. A ordem pela qual as encomendas serão produzidas deverá ser a ordem de chegada à queue de configurações.

3 Descrição dos Objetivos do Relatório

Com este relatório pretendemos demonstrar a maneira como abordamos a modelação do sistema apresentado . Adicionalmente, apresentamos também como é que essa abordagem foi transferida para os modelos UML resultantes, assim como, com a adequada fundamentação indicando também as principais mudanças ao longo das diferentes iterações.

4 Modelo de Domínio

Após uma minuciosa análise do enunciado proposto e explorando os requisitos do sistema inerentes ao mesmo, por forma a iniciar a conceção do modelo de domínio começamos por identificar as principais entidades do sistema e tentamos compreender a forma como elas interagem entre si.

Por conseguinte, esboçamos a nossa primeira versão do modelo de domínio, apresentado na Figura 2, onde consideramos que as principais entidades do nosso sistema seriam as seguintes: Componente, Pacote, Desconto, Configuração, Encomenda, Cliente, Fábrica, Stock, Fila para Produção e Fabricante.

Começando pelo **Componente**, traduzimos para o diagrama o facto de um Componente requerer outros componentes e o facto de um componente ser incompatível com outros componentes, sendo estas relações representadas através das “self transactions” que se podem ver na Figura 2. O **Stock** refere-se a um componente, ou seja, cada componente possui um stock. A **Fábrica** gera o stock (neste caso gera muitos stocks uma vez que cada um dos componentes tem um stock próprio) e possui uma fila para produção, a fila que indica qual o próximo carro a ser produzido. A **Fila para Produção** contém várias encomendas, em lista de espera para serem produzidas. A **Encomenda** está dependente do stock dos produtos e refere-se a uma configuração. O **Cliente** gera várias configurações (uma vez que pode ter configurações guardadas) e, consequentemente, tem a possibilidade de criar encomendas. O Cliente tem ainda a opção de selecionar pacotes para a sua configuração. A **Configuração** contém vários componentes (e esses componentes podem estar presentes em várias configurações em simultâneo). O **Desconto** é aplicado a uma configuração, caso existam condições favoráveis para tal. É o **Pacote** que dá direito ou não aos descontos, e o pacote pode agregar um ou mais componentes (zero não é possível). O **Fabricante**, isto é, o dono da fábrica, gera todos os pacotes.

As cardinalidades foram adicionadas consoante a interpretação mencionada anteriormente.

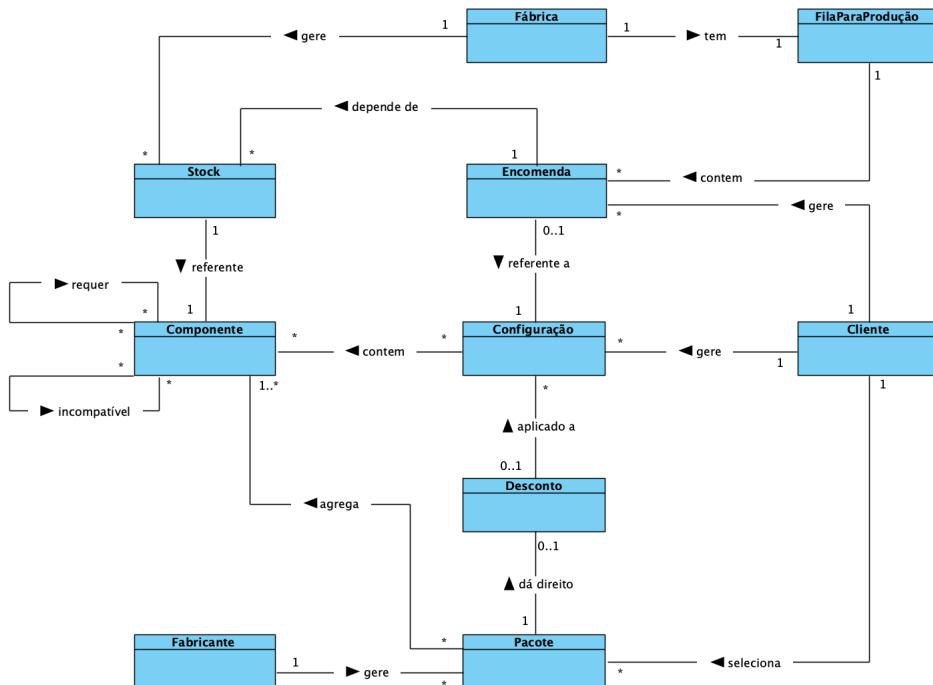


Figura 2: Modelo de Domínio v1.

Com o intuito de reduzir a complexidade do problema e adicionar mais informação útil ao nosso modelo de domínio, adicionamos uma nova entidade ao diagrama chamada **Funcionário**,

que é o Funcionário da Fábrica. Assim, tivemos que alterar o relacionamento entre a Fábrica e o Stock e, agora, são os Funcionários que atualizam o stock, sendo uma cardinalidade $* \rightarrow *$, uma vez que um Funcionário pode alterar vários stocks e um stock pode ser alterado por vários Funcionários. Fizemos ainda um outro relacionamento, onde indicamos que a Fábrica emprega vários funcionários.

O relacionamento entre o Cliente e a Encomenda foi eliminado, pois o cliente não gere encomendas, apenas expedita confirmações de configurações.

Nesta segunda versão do diagrama, apresentada na Figura 3, modificamos ainda a cardinalidade entre a Encomenda e a Configuração. Numa fase inicial consideramos que uma configuração poderia originar ou não uma encomenda, mas na verdade uma configuração pode originar também várias encomendas iguais.

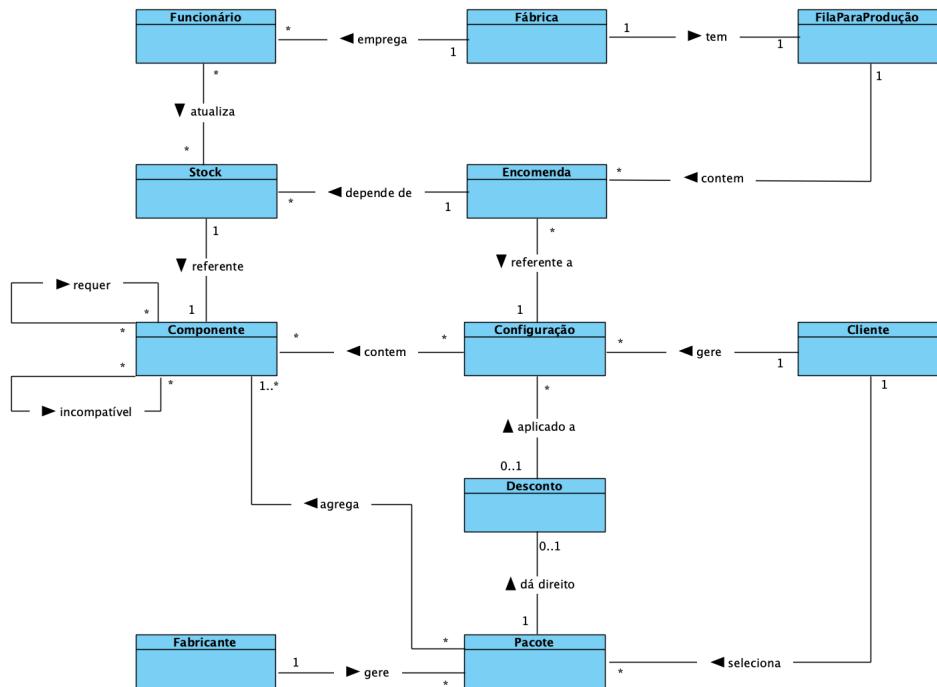


Figura 3: Modelo de Domínio v2.

Numa terceira fase sentimos a necessidade de adicionar detalhe ao nosso modelo de domínio acrescentando as entidades **Componente Básico** e **Componente Opcional**, dado que um componente é sempre classificado segundo uma destas duas generalizações.

O que nos levou a adicionar esta informação ao Modelo de Domínio foi o facto da Configuração Ótima apenas selecionar componentes opcionais; os componentes básicos já têm que estar inseridos na configuração. Deste modo, achamos importante manter sempre presente estas duas categorias de componentes para diferenciar claramente os componentes que o sistema poderá escolher na Configuração Ótima.

Em termos práticos, adicionamos uma ligação onde indicamos que um Componente Opcional “é um” Componente e que um Componente Básico “é um” Componente. Esta terceira versão do modelo de domínio encontra-se representada na Figura 4.

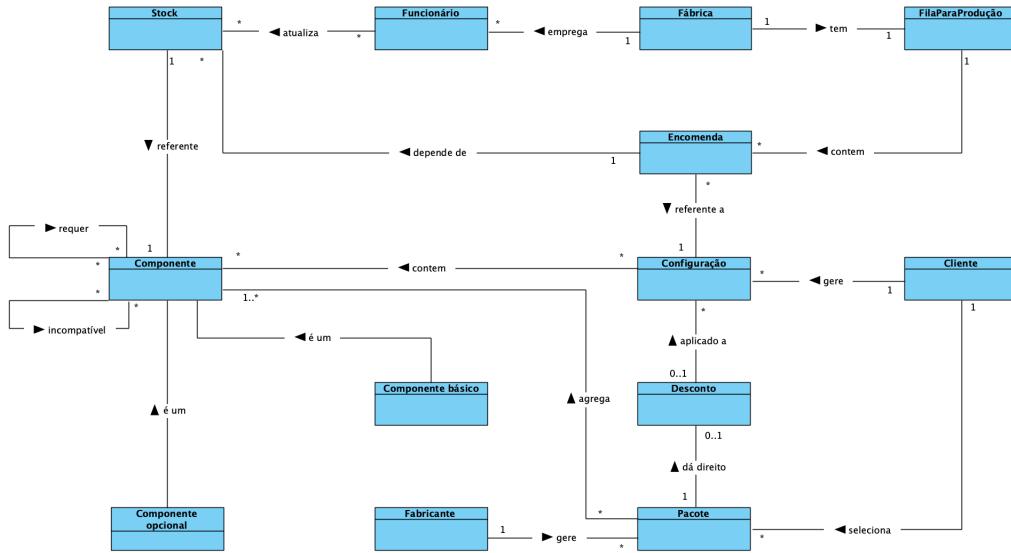


Figura 4: Modelo de Domínio v3.

Numa quarta versão do modelo de domínio, Figura 5, decidimos alterar também a cardinalidade do relacionamento Desconto-Configuração uma vez que, por exemplo, se uma configuração possuir três pacotes, cada um com um desconto diferente, a configuração acumula assim um total três descontos.

Acrescentamos uma nova entidade, o **Preço**, visto que nos nossos modelos anteriores este fator apenas estava subentendido nas ligações entre o Desconto-Pacote e Desconto-Configuração e, portanto, consideramos que ficaria mais esclarecedor adicionar um relacionamento entre o Componente e o Preço, uma vez que no enunciado está claro que todos os componentes têm um preço.

Analizando novamente a nossa solução percebemos também que a entidade Fábrica estava somente a tornar o diagrama redundante e decidimos então eliminar esta entidade, necessitando para isso de reajustar os relacionamentos, nomeadamente adicionar a relação Funcionário - Fila para Produção, como se pode ver na Figura 5.

Tendo em consideração que o Fabricante tem o papel de administrador do sistema, adicionamos também a informação de que os Fabricantes gerem os Funcionários, para assim existir a funcionalidade de serem adicionados/eliminados funcionários no sistema.

Por fim, reorganizamos as entidades e respetivas ligações no nosso diagrama para este ficar mais claro e percutível e o resultado final é o que se pode ver através da Figura 5.

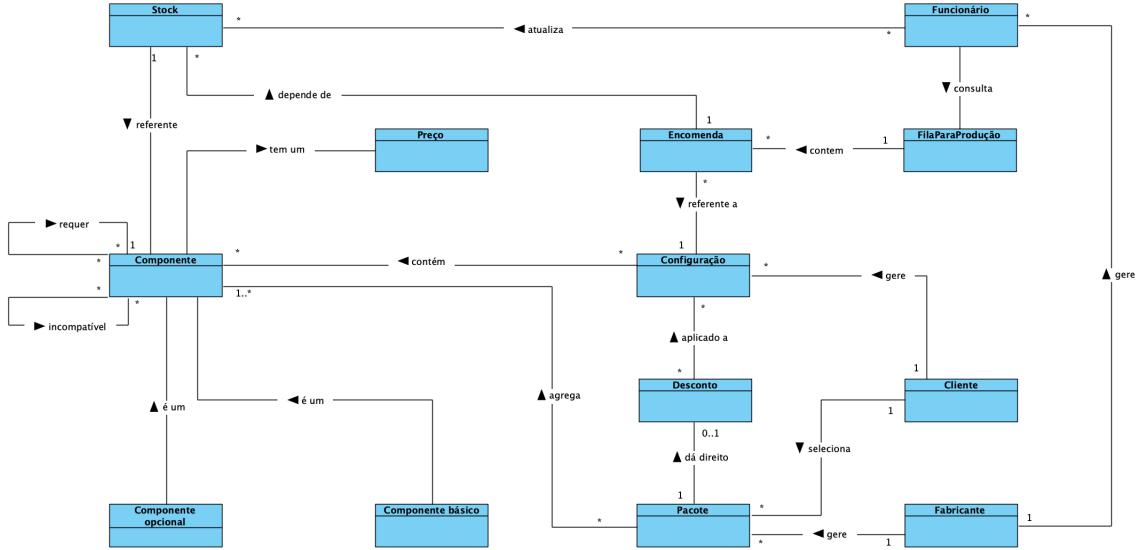


Figura 5: Modelo de Domínio v4.

Só numa última fase é que nos apercebemos que este processo era realizado no stand, com o cliente junto ao funcionário do stand. Assim sendo, concluímos que não é o cliente que seleciona os pacotes e que gera a configuração, mas sim o funcionário do stand. Para representar esta alteração no diagrama, apenas tivemos que substituir o nome da entidade Cliente para **Funcionário** uma vez que toda a logística por detrás destes conceitos funciona igualmente da forma pretendida.

Deste modo, sentimos a necessidade de especificar no modelo de domínio que o Funcionário que acrescentamos na segunda versão do modelo (Figura 3) é o **Funcionário da Fábrica**, alterando assim o nome desta entidade com o intuito de evitar futuras confusões entre ambas.

Alteramos ainda a cardinalidade do relacionamento Funcionário da Fábrica - Fila para Produção uma vez que a fila para produção pode ser consultada por vários funcionários em simultâneo.

Devido ao facto de quando a cardinalidade é 1 não é preciso especificá-la no relacionamento, eliminamos essas redundâncias e obtivemos o nosso modelo de domínio final, apresentado na Figura 6.

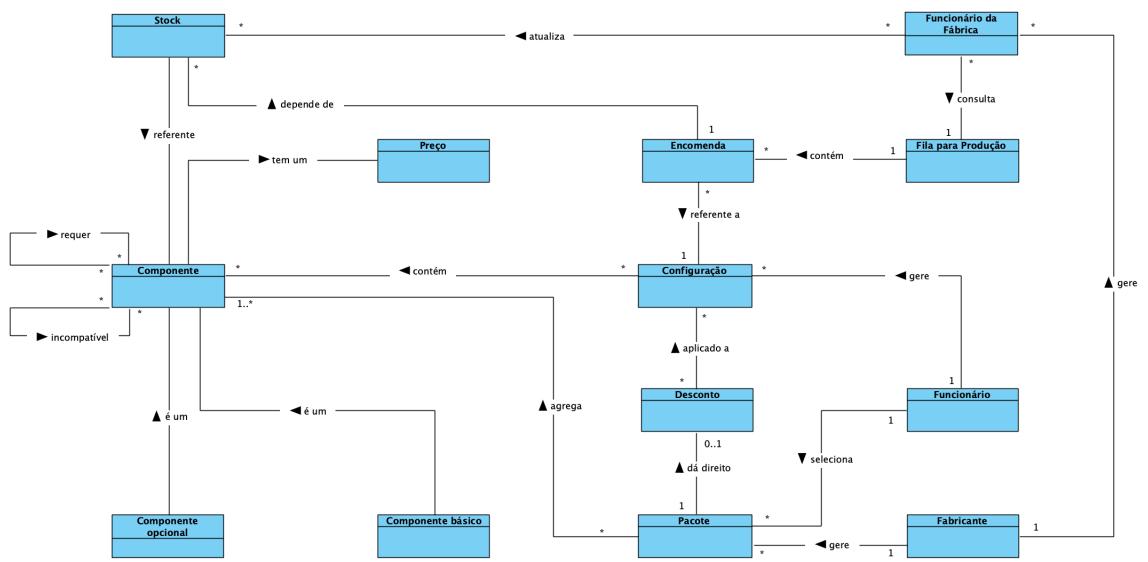


Figura 6: Modelo de Domínio Final.

5 Modelo de Use Cases

Depois de finalizado o Modelo de Domínio, chegou a vez de nos debruçarmos sobre o Modelo de Use Cases.

Dada a natureza iterativa e incremental do processo de desenvolvimento que usamos, é de salientar que estes dois modelos não foram realizados de forma sequencial, uma vez que ao longo da execução, discussão e aperfeiçoamento deste diagrama fomos sentindo a necessidade de alterar também o diagrama de domínio. É por este motivo que as primeiras versões do Modelo de Use Cases têm os mesmos nomes das entidades das primeiras versões do Modelo de Domínio.

Posto isto, a primeira tentativa de representação do Diagrama de Use Cases encontra-se na Figura 7. As entidades que consideramos serem as principais no sistema, isto é, as entidades que efetivamente vão interagir com ele são as seguintes: Cliente, Fábrica e Fabricante.

O primeiro ator que identificamos foi o Cliente. Este ator, no nosso entender, é o que tem mais funcionalidades no sistema. A primeira funcionalidade é a de **Iniciar Configuração** (especificação do Use Case na Figura 13). Dentro da configuração, o cliente tem a opção de **Adicionar Componentes** (Figura 16), **Remover Componentes** (Figura 17) e de **Adicionar Pacotes** (Figura 18). Adicionalmente, este tem ainda a opção de escolher uma **Configuração Ótima** (Figura 11), uma funcionalidade onde o cliente tem a oportunidade de ficar com a melhor configuração possível, tendo em consideração o dinheiro que ele pretende gastar. Por fim, quando a configuração estiver concluída e validada, e o cliente decidir que quer efetivamente encomendar o carro tem a opção de **Criar Encomenda** (Figura 12).

O segundo ator que detectamos foi a Fábrica. As funcionalidades da Fábrica são **Atualizar o Stock** (Figura 21), para assim poder fazer a gestão da chegada de stock de componentes à fábrica e, consequentemente, o sistema poder determinar quais são os carros que podem agora ser produzidos, **Consultar próxima encomenda com material** (Figura 19), para os respetivos funcionários da fábrica saberem qual é o próximo carro que têm que produzir e **Consultar informação do componente** (Figura 20), de modo a poderem estar informados relativamente às informações intrínsecas ao componente.

Por fim, definimos também o Fabricante como um ator. O Fabricante tem a opção de **Adicionar Novos Pacotes** (Figura 23) ao sistema, uma vez que se trata do gestor do mesmo e achamos essencial dar-lhe a opção de fazer novas agregações de componentes quando assim o entender. De igual forma, tem a alternativa de **Editar um Pacote** já existente e de **Apagar Pacotes** (Figura 22). Estas opções dão ao administrador a hipótese de escoar melhor o stock e, consequentemente, maximizar o lucro.

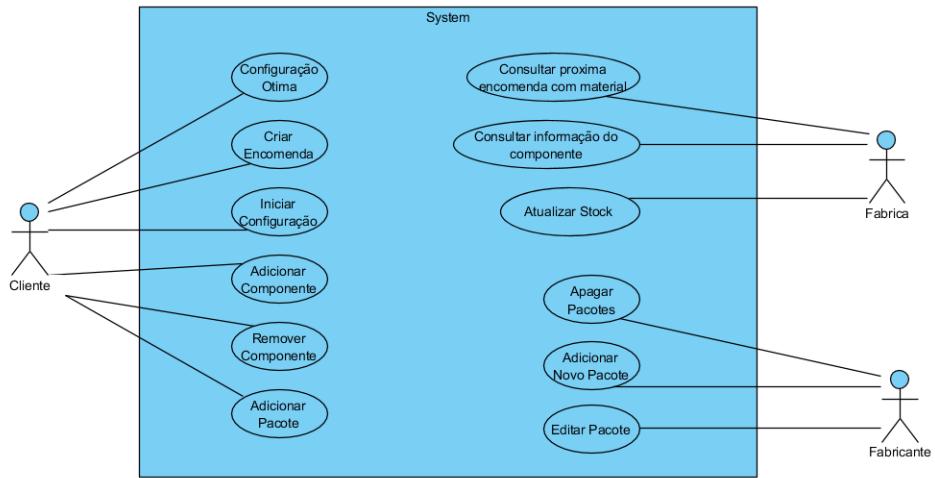


Figura 7: Modelo de Use Cases v1.

Depois de uma nova e mais profunda análise aos requisitos funcionais do sistema concluímos que o cliente deveria ter ainda a possibilidade de **Guardar Configuração** (Figura 14), por exemplo, se a iniciasse e não pretendesse finalizá-la naquele momento e, por consequência, a possibilidade de **Abrir uma Configuração** (Figura 15), para poder mais tarde retomar a configuração já guardada.

Decidimos ainda adicionar a funcionalidade **Autenticar Cliente** (Figura 25), dado que o ator deverá estar autenticado ao sistema para conseguir guardar e ter acesso às suas configurações guardadas.

Nesta fase da solução entendemos que os use cases Abrir uma Configuração e Guardar Configuração faziam parte do use case Autenticar Cliente, uma vez que para conseguirem usufruir de cada uma das duas funcionalidades mencionadas, o cliente teria que estar obrigatoriamente autenticado, ou seja, sempre que o cliente pretendesse abrir ou guardar uma configuração, ele teria também que estar autenticado. Para além disso, aquando da criação da encomenda, para os dados do cliente serem automaticamente preenchidos e a compra ser devidamente registada no nosso sistema, também consideramos que o use case Criar Encomenda incorpora o use case Autenticar Cliente.

Este novo Modelo de Use Cases encontra-se na Figura 8.

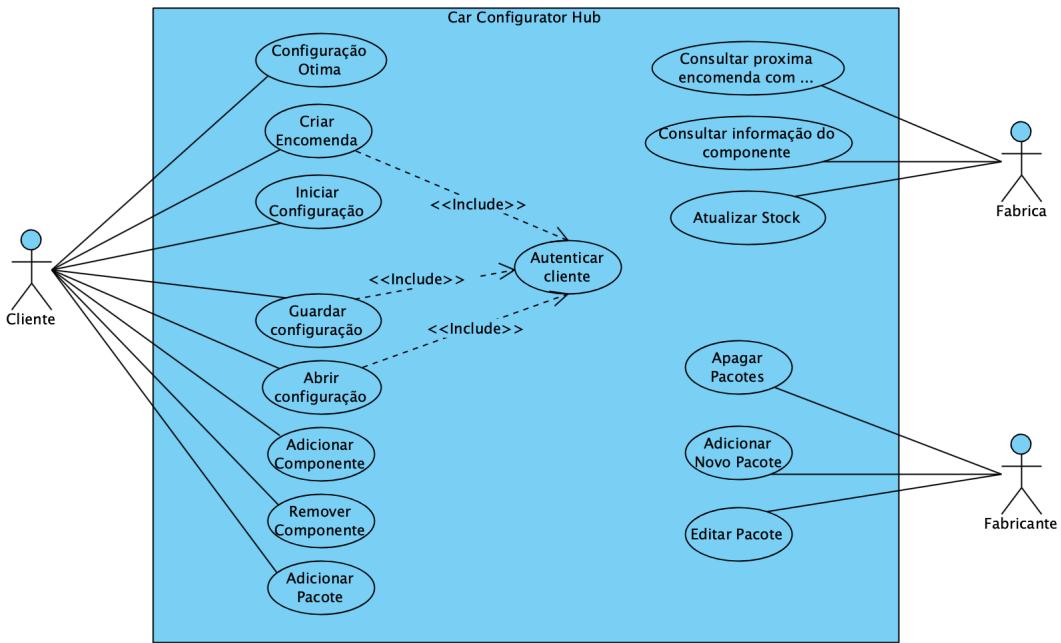


Figura 8: Modelo de Use Cases v2.

Numa terceira versão do diagrama de Use Cases, Figura 9, alteramos o nome do ator Fábrica para Funcionário, uma vez que é o próprio Funcionário da Fábrica que utiliza estas funcionalidades.

Concluímos ainda que tanto o Funcionário como o Fabricante também se podem autenticar no sistema, o primeiro para poder atualizar o stock e consultar informações, e o segundo para gerir os pacotes.

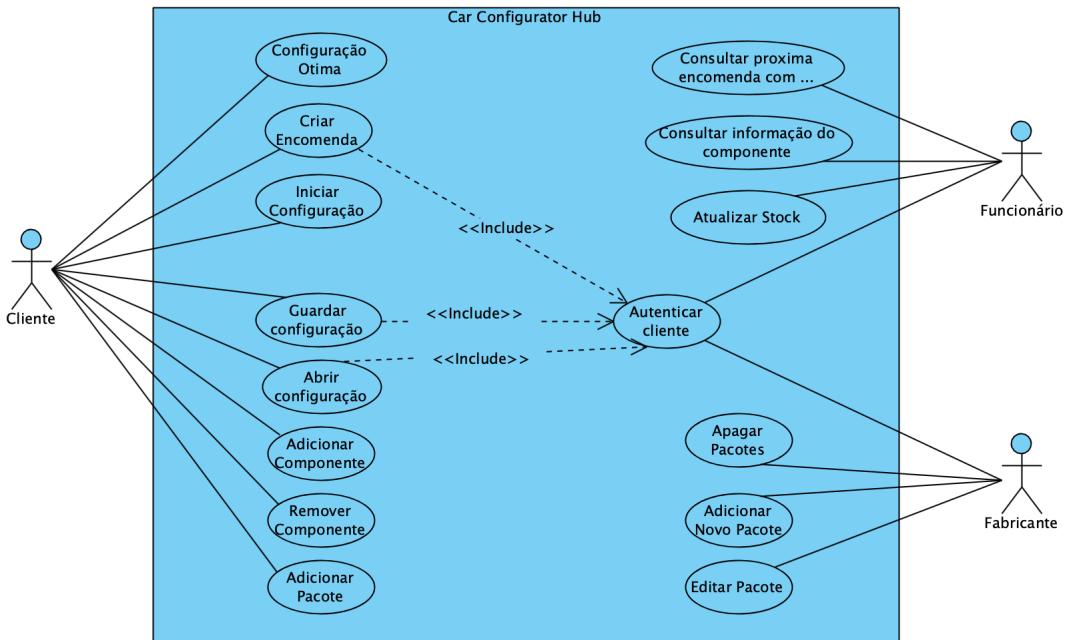


Figura 9: Modelo de Use Cases v3.

Como já foi mencionado, o nome das entidades foi sendo modificado ao longo das diferentes iterações com o intuito de obtermos um modelo mais claro, perceptível e coerente. Assim, numa nova versão do Modelo de Use Cases, alteramos o nome do Funcionário para Funcionário da Fábrica, o Cliente passou a ser o Funcionário (do stand, pelo motivo que explicamos na Secção anterior) e mudamos ainda o nome do use case “Abrir configuração” para “Abrir configuração guardada” de modo a ficar mais evidente que este use case não inicia uma nova configuração, apenas recupera uma já existente.

Decidimos também acrescentar uma associação entre o Funcionário e o Autenticar e, consequentemente, remover o include entre o Autenticar e os use cases Guardar, Abrir Configuração e Criar encomenda uma vez que agora é o Funcionário do Stand que trata desta funcionalidade e, para aceder a qualquer uma das funcionalidades do sistema, ele tem que estar sempre autenticado.

Adicionalmente, decidimos ainda remover o use case “Editar Pacote” pois tornava-se redundante o fabricante poder adicionar, apagar e editar os pacotes. Se a intenção dele for mesmo somente editar um componente de um pacote, pode na mesma efetuar esta alteração apagando o pacote e adicionando um novo, com os componentes como desejava.

Para finalizar o nosso Modelo de Use Cases faltava-nos ainda dar a opção ao fabricante de **Adicionar Funcionário** (Figura 24). Visto que o nosso objetivo é que o nosso sistema de software seja agradável e intuitivo, que irá tornar todo o workflow do stand mais rápido e apelativo para os clientes e até fazer com que os próprios funcionários do stand estejam mais motivados para persuadir os clientes na compra efetiva dos carros, as vendas vão aumentar. Inevitavelmente, o fabricante vai precisar de contratar mais funcionários e, consequentemente, adicioná-los ao sistema.

O Modelo de Use Cases final encontra-se representado na Figura 10.

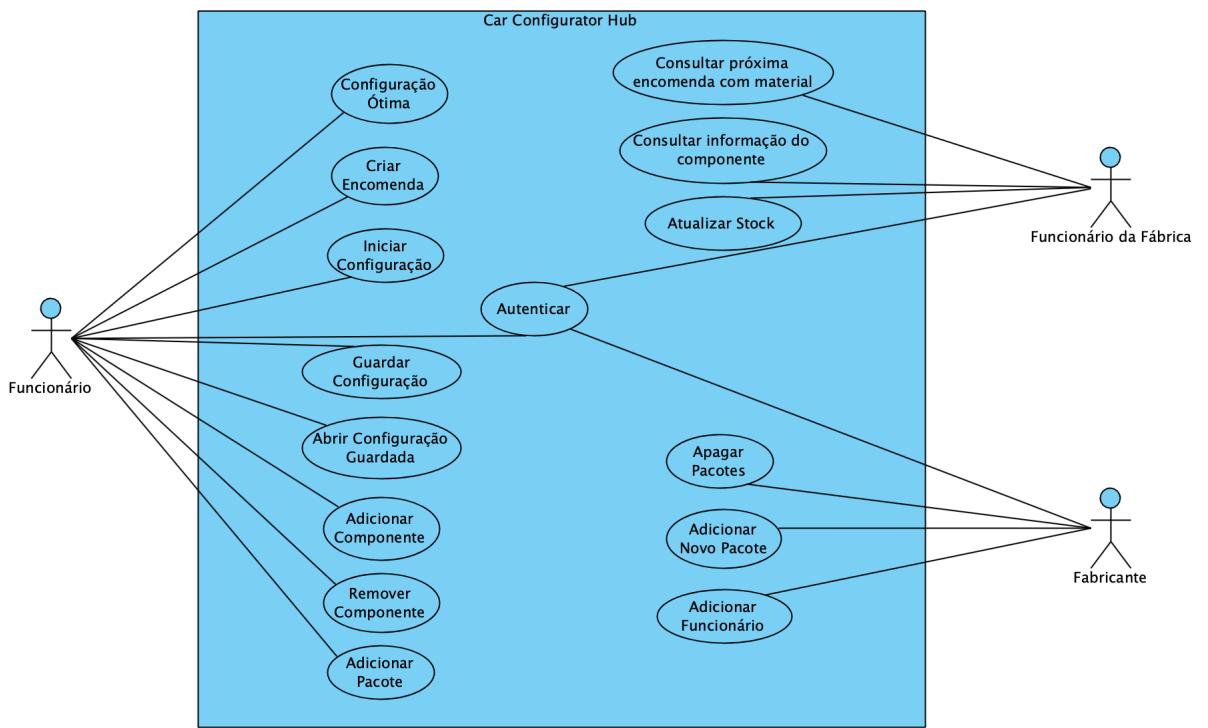


Figura 10: Modelo de Use Cases Final.

6 Especificação de Use Cases

Iremos agora proceder à especificação de cada um dos use cases presentes no Modelo de Use Cases (Figura 10).

6.1 Configuração Ótima

Use Case		
Use Case:	Configuração Ótima	
Autor:	Funcionário	
Pré-condição:	O ator tem configuração ativa com componentes básicos.	
Pós-condição:	A configuração foi atualizada.	
	Autor	Sistema
Cenário Normal	1. Indica valor máximo	
		2. Calcula a melhor configuração
		3. Propõe a nova configuração
	4. Aceita a configuração	
		5. Atualiza configuração
		6. Mensagem de confirmação
Exceção 1 [Valor demasiado baixo] (passo 1)		1.1 Indica que o valor inserido é mais baixo que o componente mais barato
Exceção 2 [Não aceita a configuração] (passo 4)		4.1 Mensagem de erro

Figura 11: Especificação do Use Case Configuração Ótima.

Para o Funcionário do stand poder usufruir da opção **Configuração Ótima** terá que adicionar a quantidade máxima de dinheiro que o cliente está disposto a gastar. Depois disso, o sistema irá calcular e propôr a nova configuração. O valor inserido poderá ser demasiado baixo e, consequentemente, não ser possível adicionar mais componentes ao sistema (propôr uma configuração ótima).

O cliente poderá não aceitar a configuração ótima sugerida pelo sistema e preferir prosseguir com a configuração manualmente.

6.2 Criar Encomenda

Use Case		
Use Case:	Criar encomenda	
Autor:	Funcionário	
Pré-condição:	O utilizador está autenticado como funcionário e tem configuração com componentes básicos.	
Pós-condição:	Foi adicionada à fila para produção uma nova encomenda.	
		Ator
Cenário Normal		
Exceção 1 [Há componentes incompatíveis] (passo 1)		
Exceção 2 [Há componentes que precisam de complementar] (passo 1)		

Figura 12: Especificação do Use Case Criar Encomenda.

O Funcionário do Stand, assim que o cliente decidir que quer encomendar o veículo, tem a opção de **Criar Encomenda**. A validade de todos os componentes é averiguada e, caso existam componentes incompatíveis ou componentes que precisem de complementares, o funcionário não poderá prosseguir com o pedido de encomenda. Se a configuração estiver válida, é devidamente adicionada à fila para produção.

6.3 Iniciar Configuração

Use Case		
Use Case:	Iniciar configuração	
Ator:	Funcionário	
Pré-condição:	Existem componentes no sistema.	
Pós-condição:	O ator tem uma configuração ativa.	
	Autor	Sistema
Cenário Normal		1. É adicionada uma nova configuração ao sistema.
		2. Mensagem de confirmação.

Figura 13: Especificação do Use Case Iniciar Configuração.

Quando um novo cliente se dirige ao stand e solicita uma configuração personalizada, o funcionário do stand irá utilizar o Car Configurator Hub para **Iniciar Configuração**. O sistema fica agora com a informação de que existe uma nova configuração.

6.4 Guardar Configuração

Use Case		
Use Case:	Guardar Configuração	
Ator:	Funcionário	
Pré-condição:	Configuração iniciada.	
Pós-condição:	Configuração guardada.	
	Autor	Sistema
Cenário Normal		1. Configuração guardada

Figura 14: Especificação do Use Case Guardar Configuração.

O Funcionário do stand pode **Guardar Configuração** no sistema para mais tarde a retomar.

6.5 Abrir Configuração Guardada

Use Case		
Use Case:	Abrir configuração guardada	
Autor:	Funcionário	
Pré-condição:	O utilizador está autenticado como funcionário e contém configurações guardadas.	
Pós-condição:	Configuração aberta.	
	Ator	Sistema
Cenário Normal		1. Apresenta as configurações guardadas
	2. Seleciona a configuração	
		3. Abre a configuração

Figura 15: Especificação do Use Case Abrir Configuração Guardada.

Para poder retomar uma configuração guardada no sistema, o funcionário do stand deverá revelar essa sua intenção e o sistema apresentará a lista das configurações existentes. O sistema expõe a configuração selecionada pelo utilizador.

6.6 Adicionar Componente

Use Case		
Use Case:	Adicionar componente	
Autor:	Funcionário	
Pré-condição:	Configuração iniciada.	
Pós-condição:	Componente adicionado.	
		Sistema
Cenário Normal		
	1. Apresenta a listagem de componentes	
	2. Seleciona Componente	
	3. Verifica dependências	
	4. Adiciona Componente	
		5. Preço da configuração é atualizado
Comp. Alternativos [Componente requer outros componentes] (passo 3)		
	3.1 Avisa utilizador que o componente requer outros componentes	
	3.2 Pergunta se quer adicionar o componente na mesma	
	3.3 Confirma	
		Regressa a 4
Comp. Alternativos [Componente Incompatível com outros componentes] (passo 3)		
	3.1.1 Avisa utilizador que o componente é incompatível com outro componente	
	3.1.2 Pergunta se quer adicionar o componente na mesma	
	3.1.3 Confirma	
		Regressa a 4
Exceção 1 [Utilizador rejeita] (passo 3.3) (passo 3.1.3)	3.3.1 Não adiciona componente	
Comp. Alternativos [Componente adicionado acabou de formar um pacote] (passo 2)	2.2.1 Informa que o cliente adquiriu um pacote	
	Regressa a 3	
Comp. Alternativos [Pacote contém desconto] (passo 2)	2.3.1 Indica que pacote originou um desconto	
	Regressa a 3	

Figura 16: Especificação do Use Case Adicionar Componente.

Se a intenção do Funcionário do stand for **Adicionar Componente** à sua configuração ele deverá escolher o componente pretendido e o sistema irá verificar as suas dependências. No caso de tudo correr como previsto, isto é, a configuração continuar válida após o componente ser inserido, o componente é efetivamente adicionado à configuração e o preço da mesma é atualizado. Porém, o componente adicionado poderá gerar incompatibilidades com os componentes presentes na configuração ou poderá requerer outros componentes. Nestes casos, o sistema irá avisar o utilizador do sucedido mas a decisão de prosseguir ou não com a adição deste componente é exclusivamente do cliente. Caso o cliente assim o decida, pode cancelar a ação.

Há a hipótese de ser formado um pacote automaticamente aquando da adição do componente e, neste caso, o cliente será informado do ocorrido e o respetivo desconto incluído (se aplicável).

6.7 Remover Componente

Use Case		
Use Case:	Remover Componente	
Ator:	Funcionário	
Pré-condição:	Configuração iniciada e existem componentes nesta.	
Pós-condição:	Componente removido.	
	Ator	Sistema
Cenário Normal	1. Indica qual dos componentes da configuração quer remover	
		2. Verifica dependências
		3. Remove componente da configuração
		4. Atualiza o preço
Comp. Alternativos [Componente pertence a pacote] (passo 2)		2.1 Avisa que componente pertence a pacote
		2.2 Pergunta se quer remover
	2.3 Confirma	
		2.4 Remove Pacote
		Retorna a 3
Comp. Alternativos [Componente selecionado é complementar de outros componentes na configuração] (passo 2)		2.1 Avisa utilizador que o componente selecionado é complementar de outros componentes
		2.2 Pergunta se quer remover o componente na mesma
	2.3 Confirma	
		Regressa a 3
Exceção 1 [Utilizador rejeita] (passo 2.3) (passo 2.1.3)		2.3.1 Não remove componente

Figura 17: Especificação do Use Case Remover Componente.

Caso o cliente pretenda **Remover Componente** da sua configuração, o sistema irá verificar as dependências e informar respetivamente o funcionário do stand. Se o componente for complementar de outro componente, o cliente irá ser questionado se pretende prosseguir com a ação. Caso o componente pertença a um pacote, o utilizador também irá ser informado que poderá perder um desconto associado. De qualquer das formas, se o componente for eliminado, o preço da configuração irá ser atualizado.

6.8 Adicionar Pacote

Use Case		
Use Case:	Adicionar pacote	
Autor:	Funcionário	
Pré-condição:	Configuração iniciada	
Pós-condição:	Pacote Adicionado	
		Ator
Cenário Normal		
	1. Apresenta a listagem dos pacotes	
	2. Seleciona pacote	
Comp. Alternativos [O pacote selecionado gera incompatibilidades na configuração] (passo 2)		
	3. Adiciona Pacote	
	4. Atualiza o preço	
Comp. Alternativos [O pacote selecionado requer outros componentes] (passo 2)		
	2.1 Avisa que o pacote selecionado gera incompatibilidades na configuração	
	2.2 Pergunta se quer adicionar o pacote mesmo assim	
Exceção 1 [Utilizador rejeita] (passo 2.3)		
	2.3 Confirma	
	Regressa a 3	
	2.1 Avisa que o pacote selecionado requer outros componentes	
	2.2 Pergunta se quer adicionar o pacote mesmo assim	
	2.3 Confirma	
	Regressa a 3	
		2.3.1 Não adiciona pacote

Figura 18: Especificação do Use Case Adicionar Pacote.

Se o cliente decidir adicionar à configuração um pacote predefinido existe a possibilidade desse pacote gerar incompatibilidades na configuração ou requerer outros componentes. Em ambos os casos, o cliente, junto do funcionário do stand, pode decidir se ainda assim pretende adicionar o pacote.

6.9 Consultar próxima encomenda com material

Use Case		
Use Case:	Consultar próxima encomenda com material	
Autor:	Funcionário da Fábrica	
Pré-condição:	Existem encomendas na fila para produção e o ator está autenticado como funcionário da fábrica.	
Pós-condição:	Uma encomenda saiu da fila para produção.	
	Ator	Sistema
Cenário Normal		1. Determina a próxima encomenda a ser produzida
		2. Remove a encomenda da fila para produção
		3. Regista funcionário como responsável pela encomenda e regista a hora e data
		4. Lista os componentes necessários para a encomenda
		5. Remove os componentes do inventário
Exceção 1 [Não existe nenhuma encomenda na fila para produção que não necessite de componentes indisponíveis] (passo 1)		1.1 Indica que não existe nenhuma encomenda que possa ser produzida.

Figura 19: Especificação do Use Case Consultar próxima encomenda com material.

O Funcionário da Fábrica poderá querer consultar qual é a próxima encomenda que deve começar a produzir. O sistema irá informar o funcionário qual a encomenda com todos os componentes que a constituem disponíveis em stock e irá atualizar devidamente o mesmo, remover a encomenda da fila para produção e registar no sistema quem é o responsável pela encomenda, assim como a respetiva hora e data da consulta. No caso de não existir componentes no inventário capazes de formar uma encomenda pronta a produzir, o funcionário da fábrica será informado.

6.10 Consultar informação do componente

Use Case		
Use Case:	Consultar informações de um componente	
Ator:	Funcionário da Fábrica	
Pré-condição:	Existem componentes registados no sistema e o ator está autenticado como funcionário da fábrica.	
Pós-condição:	É registada a hora e a data da consulta no sistema.	
	Ator	Sistema
Cenário Normal	1. Indica o componente que pretende consultar	
		2. Verifica a existência do componente indicado
		3. Regista a data e hora da consulta e o funcionário que a executou
		4. Mostra informações relativas ao componente
Exceção 1 [O componente não existe no sistema] (passo 2)		2.1. Indica que o componente não existe no sistema

Figura 20: Especificação do Use Case Consultar informação do componente.

Existe a opção de serem consultadas as informações relativas a um componente. Neste caso, o funcionário da fábrica indica qual o componente que pretende consultar e são lhe apresentadas as respetivas informações. Caso o componente não exista no sistema, o utilizador é avisado.

6.11 Atualizar Stock

Use Case		
Use Case:	Atualizar stock de componentes	
Ator:	Funcionário da Fábrica	
Pré-condição:	Ator está autenticado como funcionário da fábrica	
Pós-condição:	O stock referente aos componentes foram atualizados	
	Ator	Sistema
Cenário Normal	1. Indica componentes a alterar	
	2. Indica quantidades	
		2. Atualiza o stock dos componentes indicados
		3. Determina as encomendas que podem agora ser produzidas
		5. Apresenta uma mensagem de sucesso
Exceção 1 [Componente não existe] (passo 1)		1.1 Informa que não existe um componente no sistema
Exceção 2 [Quantidade inválida] (passo 2)		2.1 Informa que a quantidade inserida é inválida (menor que 0)

Figura 21: Especificação do Use Case Atualizar Stock.

Quando chega nova mercadoria à fábrica ou a presente é danificada, o funcionário da fábrica é responsável por atualizar o respetivo stock no sistema. Assim, ele deverá indicar quais os componentes a alterar e as suas quantidades. O sistema irá determinar quais são os carros que podem ser produzidos com esta alteração e irá registar a data e hora da atualização, assim como o funcionário que a executou. Caso algum dos componentes mencionados pelo funcionário da fábrica não exista no sistema, o seu stock não é atualizado.

6.12 Apagar Pacotes

Use Case		
Use Case:	Apagar pacotes	
Autor:	Fabricante	
Pré-condição:	Existe pelo menos um pacote no sistema	
Pós-condição:	Pacote removido	
	Ator	Sistema
Cenário Normal	1. Indica o pacote que pretende apagar	
		2. Remove pacote
Exceção 1 [Não existe o pacote no sistema] (passo 1)		1.1 Informa que pacote não existe

Figura 22: Especificação do Use Case Apagar Pacotes.

O administrador do sistema tem a possibilidade de **Apagar Pacotes** do sistema. Caso ele pretenda eliminar um pacote que não exista, o utilizador é informado que o pacote não existe.

6.13 Adicionar Novo Pacote

Use Case		
Use Case:	Adicionar novo pacote	
Ator:	Fabricante	
Pré-condição:	Existem componentes registados no sistema	
Pós-condição:	Novo pacote no sistema	
	Ator	Sistema
Cenário Normal	1. Indica o nome e o desconto total sobre o pacote	
	2. Indica quais são os componentes que constituem o pacote	
		3. Valida que os componentes são todos compatíveis
		4. Novo pacote adicionado ao sistema
Exceção 1 [Um dos componentes não existe] (passo 1)		1.1 Informa que componente não existe no sistema
Comp. Alternativos [Componentes incompatíveis] (passo 3)		3.1 Informa que os componentes não são compatíveis
		3.2 Informa que o pacote não foi registado

Figura 23: Especificação do Use Case Adicionar Novo Pacote.

O Fabricante pode **Adicionar Novo Pacote** ao sistema. Para tal, terá que indicar o nome e o desconto total aplicado sobre o pacote e os componentes que o constituem. Se algum dos componentes não existir, o pacote não é registado. Se o fabricante selecionar componentes que não são compatíveis, o pacote também não é registado.

6.14 Adicionar Funcionário

Use Case		
Use Case:	Adicionar Funcionário	
Ator:	Fabricante	
Pré-condição:	O utilizador está autenticado como fabricante	
Pós-condição:	Funcionário adicionado	
	Ator	Sistema
Cenário Normal	1. Indica os dados do funcionário a adicionar, podendo se tratar de um funcionário da fábrica ou do stand	
		2. Verifica se os dados são únicos
		3. Regista funcionário
Exceção 1 [Dados não são únicos] (passo 2)		2.1 Informa que os dados não são únicos
		2.2 Informa a impossibilidade de registo

Figura 24: Especificação do Use Case Adicionar Funcionário.

Uma outra funcionalidade do Fabricante é a de adicionar funcionários tanto da fábrica como do stand ao sistema. Para isso, basta inserir os dados do novo funcionário, que serão posteriormente validados pelo sistema. No caso de já existirem essas credenciais, o funcionário não é registado. Caso contrário, o funcionário é adicionado com sucesso.

6.15 Autenticar

Use Case		
Use Case:	Autenticar	
Ator:	Funcionário/Fabricante/Funcionário da Fábrica	
Pré-condição:	O ator não está autenticado	
Pós-condição:	O ator ficou autenticado	
	Ator	Sistema
Cenário Normal	1. Insere dados de acesso	
		2. Valida credenciais
		3. Informa que está autenticado
Exceção 1 [Credenciais inválidas] (passo 2)		1. Informa que as credenciais são inválidas

Figura 25: Especificação do Use Case Autenticar.

Tanto o Funcionário, como o Fabricante e o Funcionário da Fábrica têm que se autenticar no sistema. Após a tentativa de acesso, as suas credenciais são validadas. Se as credenciais inseridas forem inválidas, o utilizador é advertido.

7 Idealização do Modelo da Interface com o Utilizador da 1.^a Fase do Projeto

7.1 Máquina de estado

Para abordar a modelação do comportamento interface, tendo como objetivo compreender melhor o aspeto global desta, decidimos criar um diagrama de máquinas de estado. Como a metodologia que, na nossa opinião, representa a forma mais natural de compreensão de um sistema complexo é a de o decompor hierarquicamente, dado o elevado poder de descrição que o modelo resultante adquire, decidimos aplicá-la.

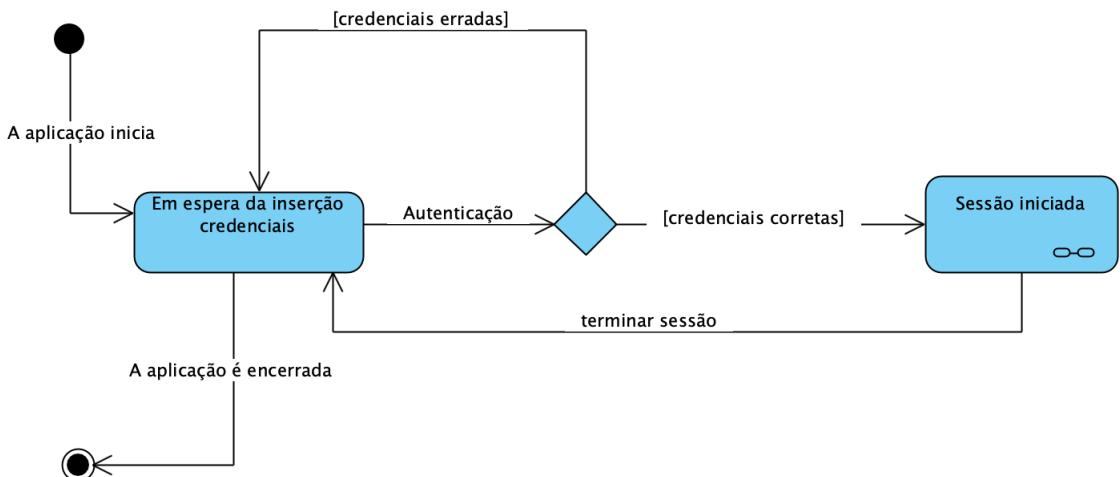


Figura 26: Modelo Máquina de Estado do Sistema

Os principais estados que a interface do utilizador terá são, como mostra a Figura 26, o estado **Sessão iniciada** e **Em espera da inserção de credenciais**. Deste último a transição que leva a interface a **Sessão iniciada** é a de autenticação. Esta transição, é apenas válida quando as credenciais estão corretas, caso contrário, a interface mantém-se no estado de **Em espera da inserção de credenciais**. Do estado **Sessão iniciada**, apenas há uma transição que, designada por terminar sessão, é encadeada quando o utilizador termina a sessão e leva a interface para o estado **Em espera da inserção de credenciais**.

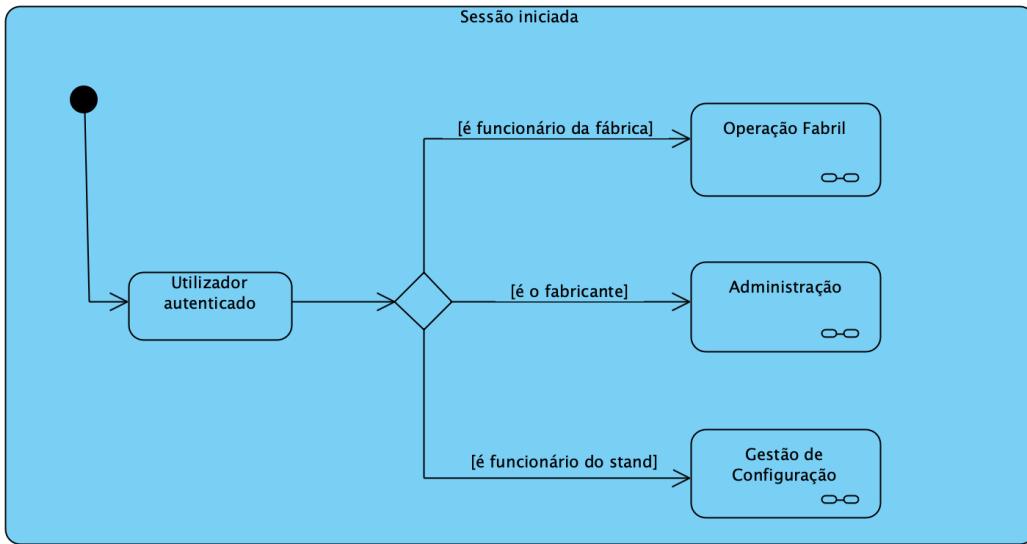


Figura 27: Expansão do Estado Composto

Após esboçado o comportamento dos estados já referidos, decidimos elaborar os constituintes do estado **Sessão iniciada** promovendo este a estado composto. Na Figura 27 é possível verificar o resultado desta elaboração. O estado **Sessão iniciada** é também ele composto por sub-estados, mais precisamente, pelo sub-estado **Utilizador autenticado**, que é o estado inicial de sessão iniciada, o sub-estado **Operação Fabril**, que se destina a utilizadores que são funcionários da fábrica, o sub-estado **Administração** que se destina ao utilizador fabricante e por fim o sub-estado **Gestão de Configuração** que é o estado destinado aos funcionários do stand.

Para que as interações da interface com o utilizador fossem mais evidentes e por forma a simplificar a compreensão desta, consideramos que o melhor passo a seguir seria o de aumentar a complexidade do modelo nos sub-estados **Operação Fabril**, **Administração** e **Gestão de Configuração**. Adicionalmente, nos estados mencionados, dada a sua semelhança comportamental, decidimos também atribuir a mesma arquitetura à máquina de estado interna. Mais concretamente, decidimos que estes teriam como sub-estado inicial um **Centro de controlo**. Este **Centro de controlo** é responsável por manter uma visão geral das possibilidades do super-estado correspondente.

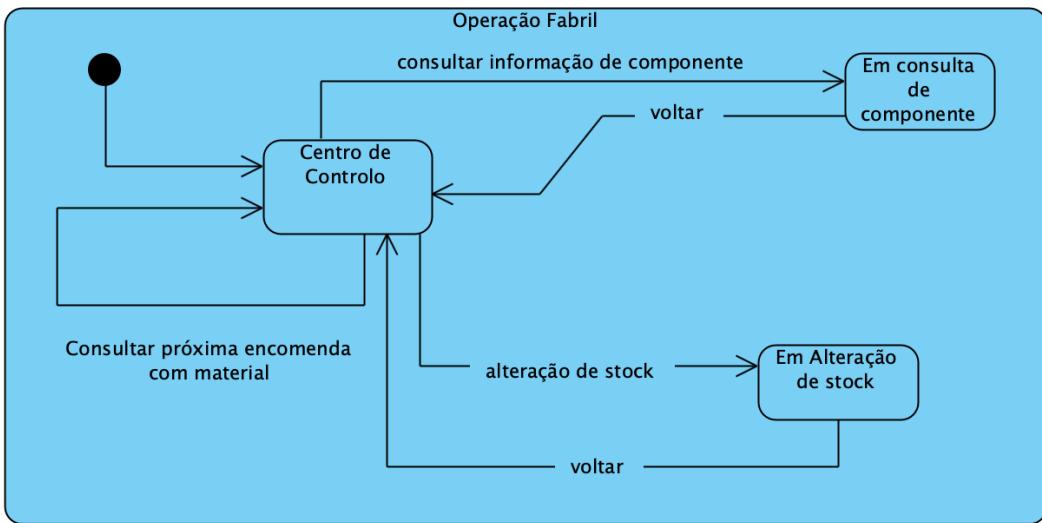


Figura 28: Expansão do Estado Composto

Em **Operação Fabril** (Figura 28), a partir do sub-estado **Centro de controlo**, é possível consultar a próxima encomenda com material e, adicionalmente, pode-se também transitar para o sub-estado **Em alteração de stock** ou para o sub-estado **Em consulta de componente**. O sub-estado **Em alteração de stock** é onde é alterado o stock referente a um componente. De forma semelhante, o estado **Em consulta de componente** é onde são visualizadas as propriedades de manufatura, assim como, a quantidade de stock de um determinado componente.

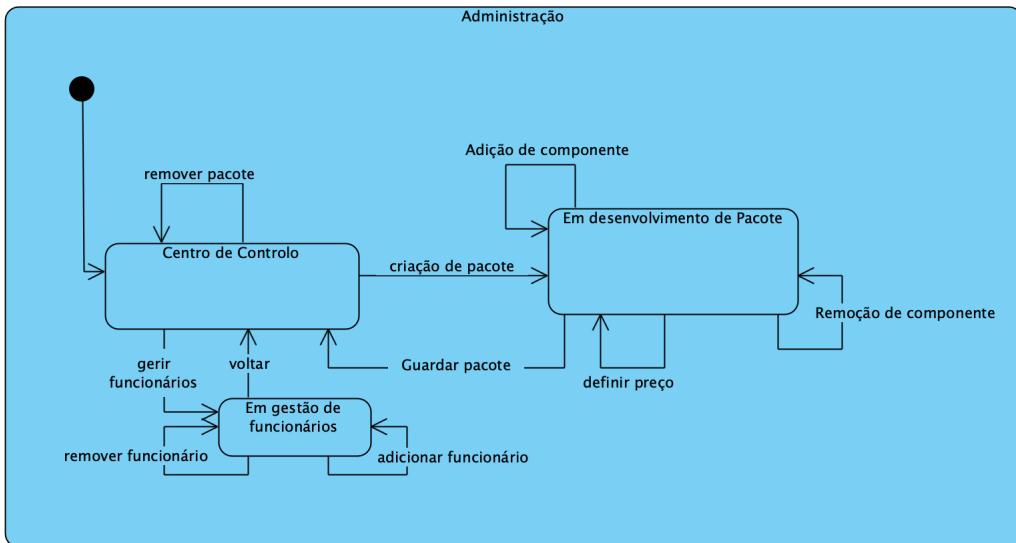


Figura 29: Expansão do Estado Composto

Em **Administração** (Figura 29), a partir do sub-estado **Centro de controlo**, é possível transitar para um sub-estado **Em gestão de funcionários** assim como para o sub-estado **Em desenvolvimento de Pacote**. O sub-estado **Em gestão de funcionários** permite a adição e remoção de funcionário tanto da fábrica como do stand. O sub-estado **Em desenvolvimento de Pacote**, tal como nome indica, é destinado à criação e gestão de um pacote. Esta gestão é feita através da adição e remoção de componentes, assim como, a atribuição de um desconto por meio

da definição de um preço.

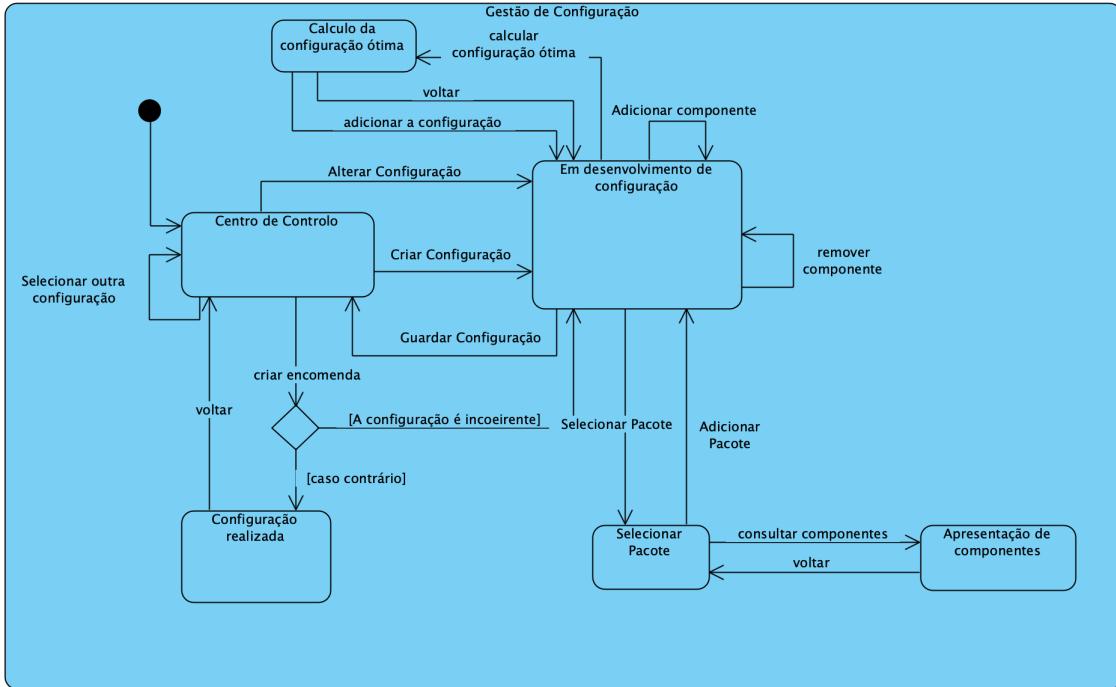


Figura 30: Expansão do Estado Composto

Em **Gestão de Configuração** (Figura 30), ao sub-estado **Centro de controlo** está associada uma configuração que é possível ser trocada usando a auto-transição própria para esse efeito. Este mecanismo, permite que quando a interface transita para o sub-estado **Configuração realizada** apenas a configuração selecionada é expedida, quando esta não contém configurações que requerem outros componentes ou componentes incompatíveis. A criação de novas configurações ou de alteração das existentes, partindo de **Centro de Controlo**, é feita no sub-estado **Em desenvolvimento de configuração**. Este sub-estado permite adicionar e remover componentes, assim como transitar para o sub-estado **Selecionar Pacote**, onde podem ser adicionados pacotes. Para além do já mencionado, o sub-estado **Em desenvolvimento de configuração** permite a transição para o sub-estado **Cálculo da configuração ótima** assim como o retorno para **Centro de Controlo** ao guardar a configuração.

7.2 Prototipagem da UI

Após completar o diagrama de máquina de estados, começamos a desenvolver o protótipo da interface de utilizador em conformidade com as decisões previamente tomadas. As principais preocupações foram tentar criar um interface intuitivo, de uso fácil e rápido. Como a utilização do software será feita pelos funcionários do stand não priorizamos ter uma interface bonita mas simples, clara que permita realizar a configuração do veículo o mais rápida e eficazmente possível.

1 - Sem sessão iniciada:

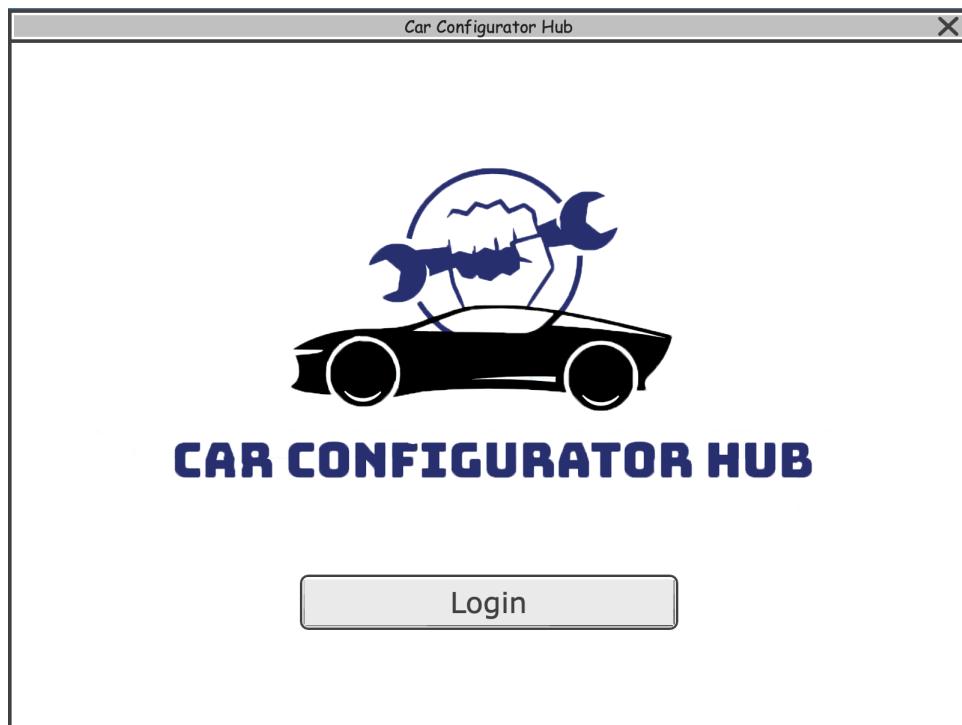


Figura 31: Página Inicial

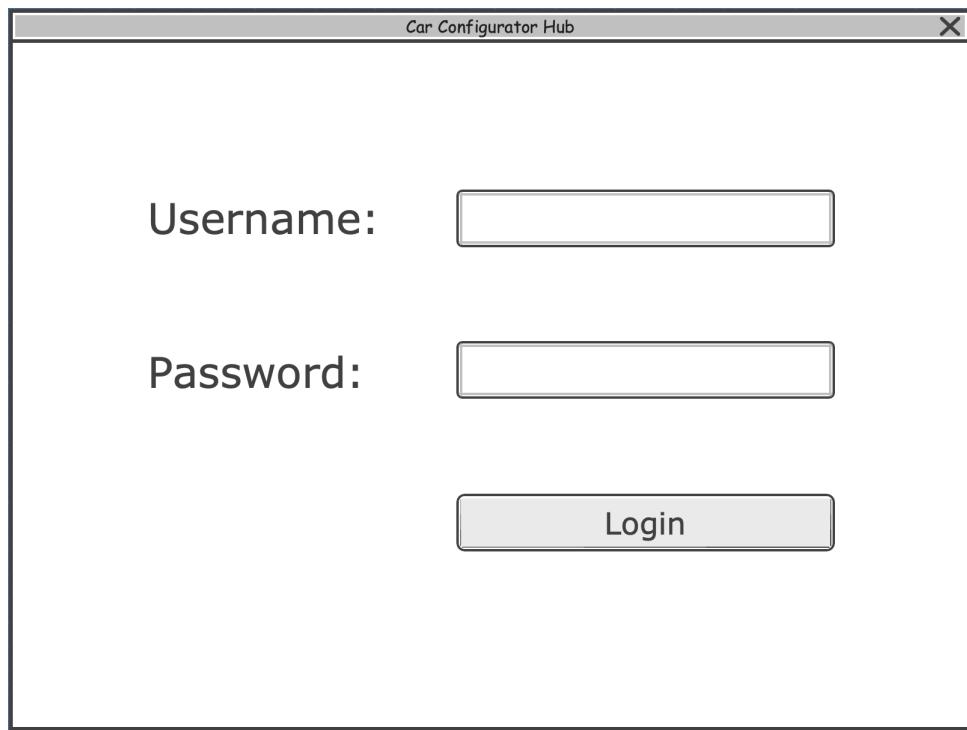


Figura 32: Em espera da inserção de credenciais

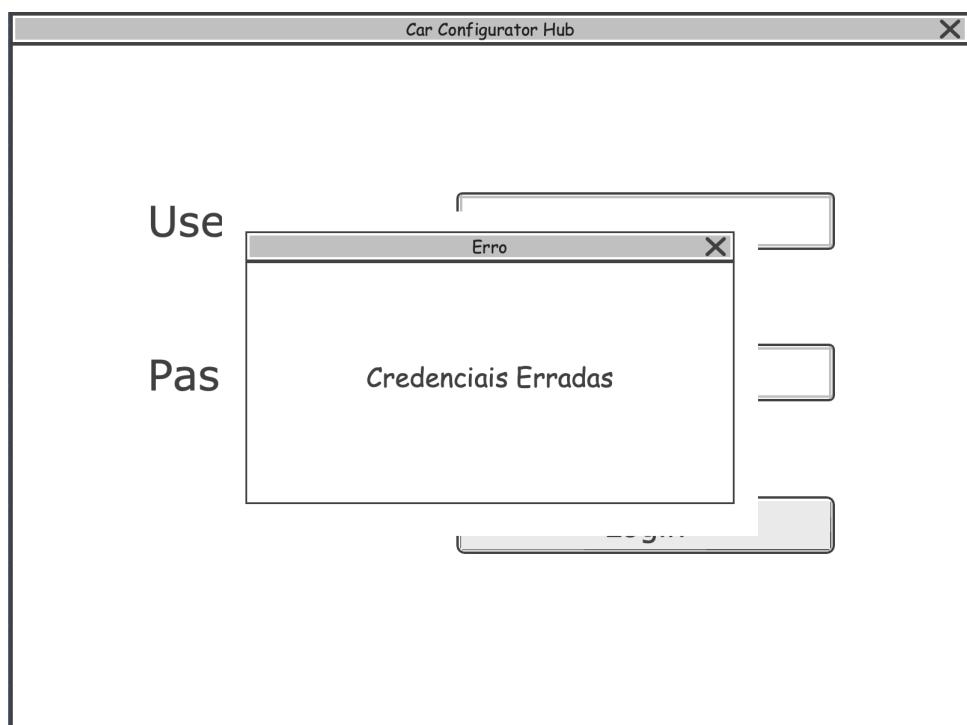


Figura 33: Erro de credenciais erradas

2 - Com sessão iniciada

Após a autenticação os utilizadores serão redirecionados para o respetivo Centro de Controlo dependendo do seu tipo, Funcionário da Fábrica, Fabricante ou Funcionário, onde estarão disponíveis as funcionalidades que cada um poderá usar.

2.1 - Funcionário da Fábrica

Os funcionários da fábrica têm apenas disponível uma janela na qual poderão executar as suas tarefas, editar o stock dos vários componentes existentes no sistema e ver qual o Id da próxima encomenda.

The screenshot shows a window titled "Car Configurator Hub". Inside, there is a table with three columns: "Componente", "Stock", and "Preço (€)". The table contains four rows of data:

Componente	Stock	Preço (€)
Componente 1	21	220
Componente 2	21	2132
Componente 3	0	32021
Componente 4	10000	21

Clique duplo para editar

Próxima Encomenda: 32143141

Concluir

Figura 34: Operação Fabril

2.2 - Administração

O Fabricante é o responsável por administrar o sistema, para isso tem 2 tarefas, a gestão de pacotes e de funcionários. Quando autenticado, o utilizador é redirecionado para o centro de controlo onde pode adicionar ou remover pacotes, abrir a janela de edição de componentes de determinado pacote ou abrir a janela de gestão de funcionários.

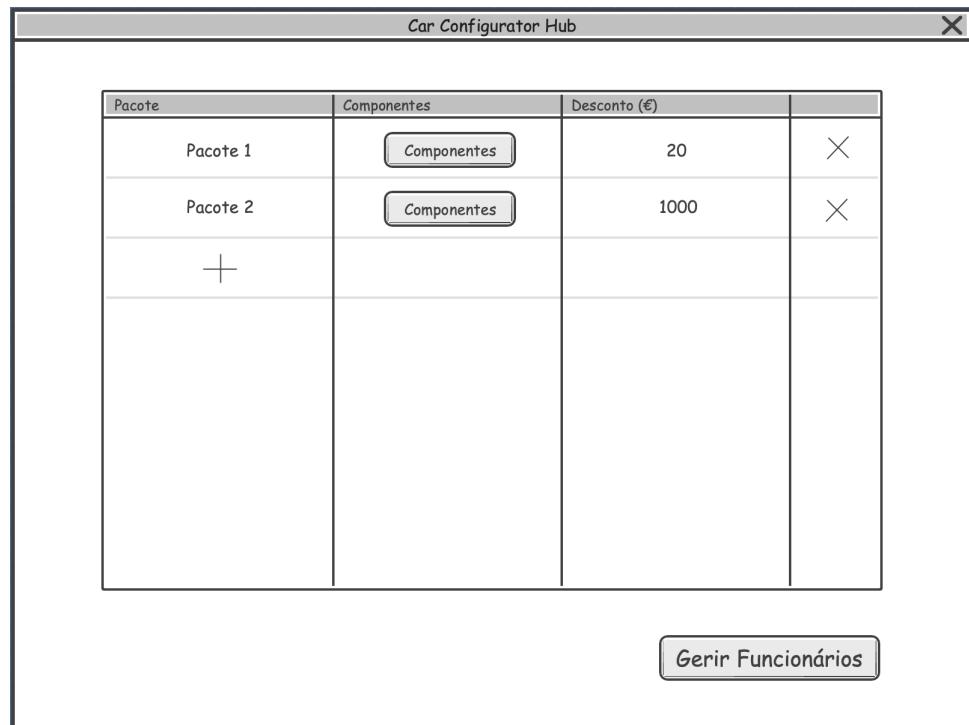


Figura 35: Centro de Controlo

2.2.1 - Desenvolvimento de Pacote

O administrador pode saber se o componente existe em stock ou não e o seu preço, remover componentes e abrir a janela de consulta de componentes disponíveis no sistema ou poderá selecionar componentes para adicionar ao pacote.

Componente	Stock	Preço (€)	
Componente 1	<input checked="" type="checkbox"/>	220	X
Componente 2	<input type="checkbox"/>	213	X
Componente 3	<input checked="" type="checkbox"/>	32021	X
Componente 4	<input checked="" type="checkbox"/>	21	X

Componentes

Figura 36: Componentes de Pacote

Componente	Stock	Preço (€)	
Componente 1	<input checked="" type="checkbox"/>	220	X
Componente 2	<input type="checkbox"/>	213	X
Componente 3	<input type="checkbox"/>	32021	X
Componente 4	<input type="checkbox"/>	21	X

Aviso

O Componente 1 será permanentemente apagado.

Pretende continuar?

Cancelar Confirmar

Componentes

Figura 37: Aviso de remoção de componente

Car Configurator Hub

A screenshot of a software window titled "Car Configurator Hub". Inside, there is a table with three columns: "Componente", "Stock", and "Preço (€)". The table contains four rows of data:

Componente	Stock	Preço (€)
Componente	21	22
Componente	21	213
Componente	0	3202
Componente	1000	21

Below the table, a message reads "Clique duplo para selecionar" (Double-click to select).

Figura 38: Seleção de novo componente

Car Configurator Hub

A screenshot of a software window titled "Car Configurator Hub". Inside, there is a table with three columns: "Componente", "Stock", and "Preço (€)". The table contains four rows of data. A modal dialog box titled "Erro" (Error) is overlaid on the table, displaying the message "Incompatível com:" followed by two component names: "Componente" and "Componente 2".

Componente	Stock	Preço (€)
Componente	21	22
Componente	21	213
Componente	0	3202
Componente	1000	21

Below the table, a message reads "Clique duplo para selecionar" (Double-click to select).

Figura 39: Erro de incompatibilidade de componentes

2.2.2 - Gestão de Funcionários

O administrador pode gerir funcionários inteiramente a partir deste ecrã, não só adicionar ou remover como também fazer qualquer tipo de edição.

User	Password	Nome	Tipo	X
1	1234	Funcionário A	Fábrica	X
2	1234	Funcionário B	Stand	X
+				

Figura 40: Janela de gestão de funcionários

User	Password	Nome	Tipo	X
1	1234	Funcionário A	Fábrica	X
2	1234	Funcionário B	Stand	X
+				

Aviso

O Funcionário A será permanentemente apagado.

Pretende continuar?

Cancelar Confirmar

Figura 41: Aviso de remoção de funcionário

2.3 - Funcionário

A única tarefa do funcionário é fazer a gestão de configurações. Na janela principal ele pode criar ou remover configurações ou abrir a janela de edição de configuração.

Pacote	Preço (€)	
Configuração 1	2004	X
Configuração 2	10020	X
+		

Figura 42: Gestão de configurações

Pacote	Preço (€)	
Configuração 1	2004	X
Configuração 2	10020	X

Aviso

A Configuração 1 será permanentemente apagada.

Pretende continuar?

Cancelar Confirmar

Figura 43: Aviso na remoção de uma configuração

2.3.1 - Edição de configuração

Nesta janela o funcionário pode ver se os componentes da configuração se encontram ou não em stock, remover componentes e abrir as janelas de seleção de componentes, seleção de pacotes, geração de configuração ótima e de criação de encomenda.



Figura 44: Gestão de uma configuração



Figura 45: Aviso de remoção de componente

2.3.1.1 - Seleção de componentes

Nesta janela o funcionário escolhe componentes para adicionar à configuração.

The screenshot shows a window titled "Car Configurator Hub". At the top center is the word "Motor" with left and right arrows on either side, indicating it's a filter or search term. Below this is a table with three columns: "Componente", "Stock", and "Preço (€)". The table contains four rows of data:

Componente	Stock	Preço (€)
Componente	21	22
Componente	21	213
Componente	0	3202
Componente	1000	21
Clique duplo para selecionar		

Figura 46: Listagem de componentes para adicionar

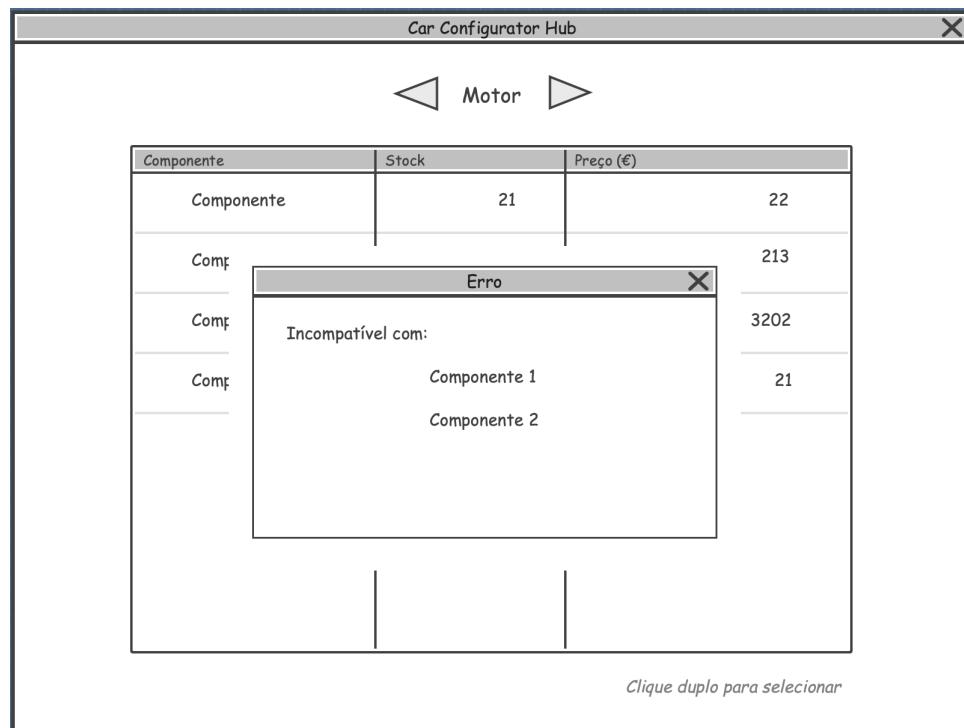


Figura 47: Aviso de incompatibilidade de componentes

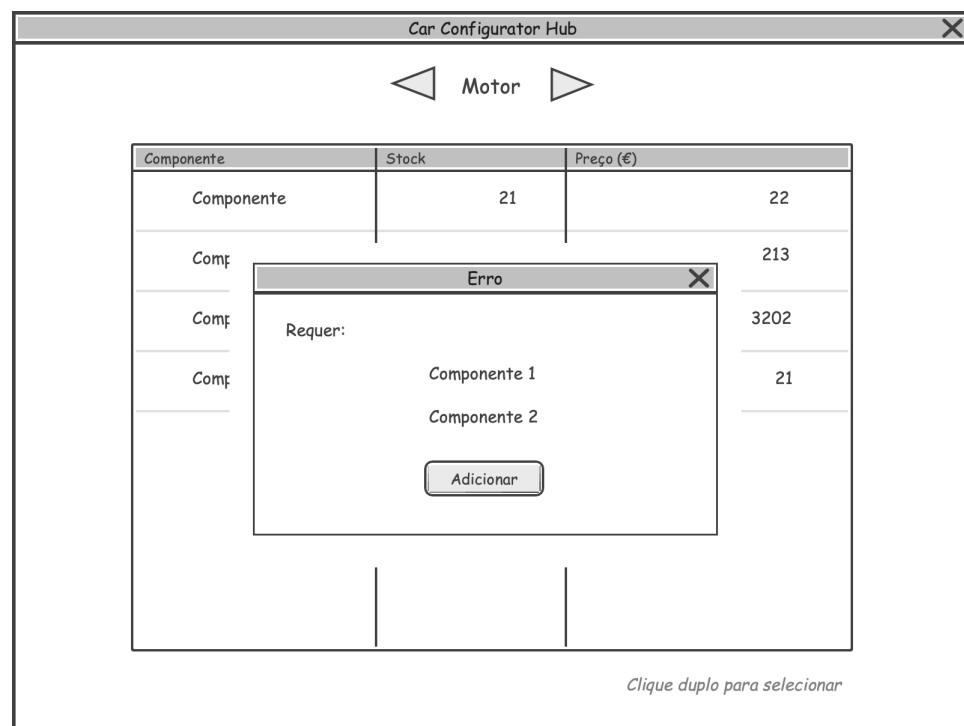


Figura 48: Aviso de necessidade de componentes

2.3.1.2 - Seleção de pacotes

Nesta janela o funcionário pode escolher pacotes de componentes para adicionar à configuração ou abrir uma janela que mostra quais os componentes de determinada configuração.

Pacote	Componentes	Desconto (€)
Pacote 1	Componentes	20
Pacote 2	Componentes	1000

Figura 49: Listagem de pacotes disponíveis

Componente	Stock	Preço (€)
Componente	21	22
Componente	21	213
Componente	0	3202
Componente	1000	21

Figura 50: Listagem de componentes de um pacote

2.3.1.2 - Geração de configuração ótima

Nesta janela, o funcionário tem a oportunidade de introduzir o valor máximo que o cliente está disposto a gastar e o sistema automaticamente gerará uma configuração ótima que depois poderá ou não ser aplicada.

Componente	Stock	Preço (€)
Componente 1	21	22
Componente 2	21	213
Componente 3	0	3202
Componente 4	1000	21

Valor máximo: 3400.23 €

Gerar Conf. Ótima Aplicar

Figura 51: Geração de configuração ótima

2.3.1.3 - Criação de encomenda

Nesta janela, depois de completa a configuração, o funcionário deverá introduzir os dados do cliente e, em seguida, submeter a encomenda para produção.

The screenshot shows a window titled "Car Configurator Hub". Inside the window, there are five input fields with labels: "Nome:", "Número de Identificação:", "Morada:", "País:", and "Email:". Below these fields is a large, light-gray rectangular button labeled "Cria Encomenda". The window has a standard title bar with a close button in the top right corner.

Figura 52: Criação de encomenda

8 Máquina de Estado da Configuração

Tendo em consideração que a Configuração é uma entidade fundamental no nosso sistema decidimos modelar a sua máquina de estados, tal como se pode ver na figura seguinte:

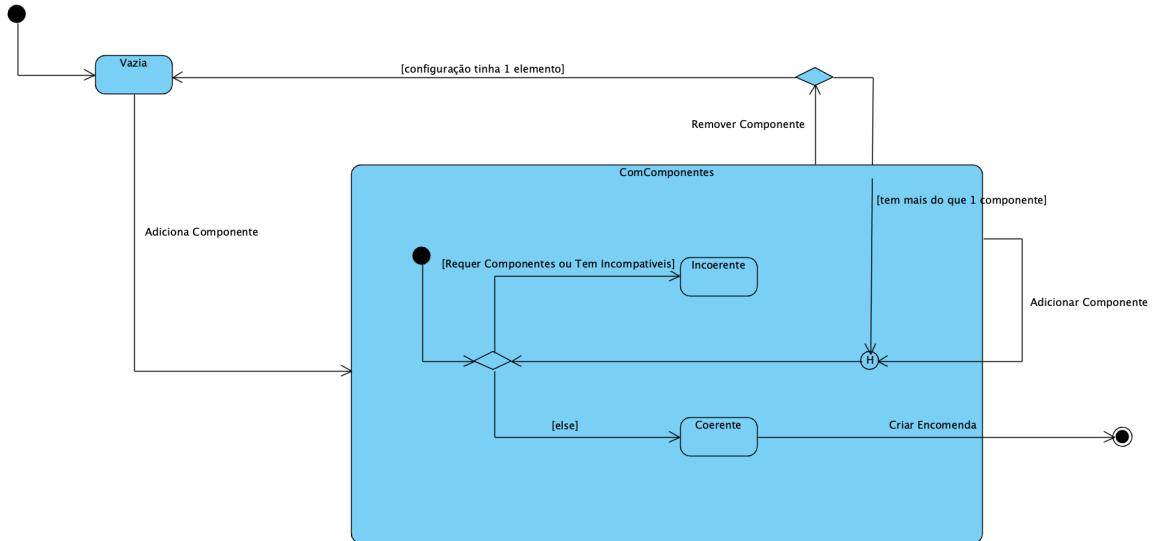


Figura 53: Máquina de Estado da Configuração.

Esta entidade poderá ter dois estados: vazia ou com componentes. Se estiver com componentes, poderá ter dois subestados: incoerente ou coerente. Estes dois subestados devem-se ao facto de quando se adiciona um componente novo à configuração, o mesmo poderá ser incompatível com algum dos componentes presentes na configuração ou poderá requerer algum componente que não esteja presente na configuração, o que leva a configuração para um estado incoerente.

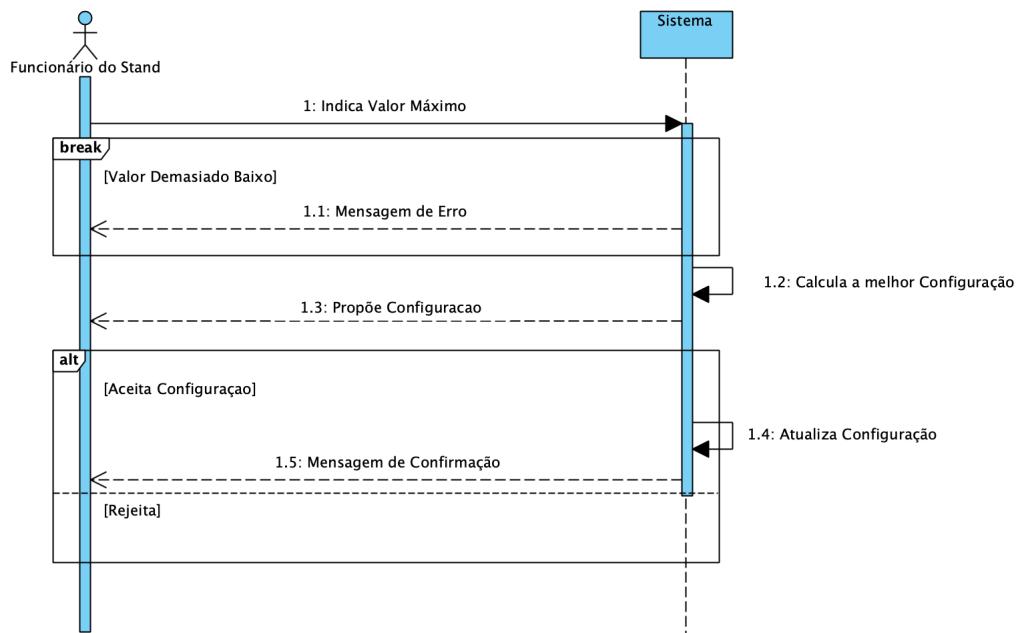
No final de todos as adições de componentes a configuração deverá estar num estado coerente uma vez que só assim é possível criar uma encomenda.

O histórico vai sendo sempre guardado para não perdermos o estado da configuração.

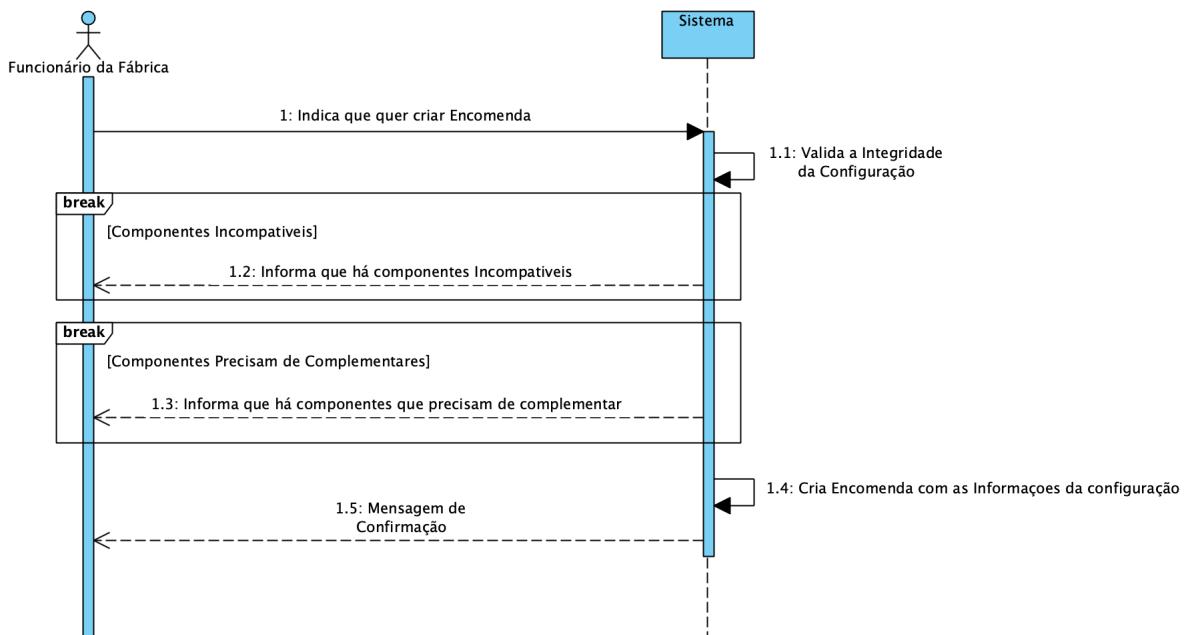
9 Diagrama de Sequência de Sistemas

Com base nos uses cases e nas respectivas especificações construímos um Diagrama de Sequência de Sistemas para cada um dos use cases já identificados. Estes diagramas permitem-nos compreender com um maior pormenor o fluxo de eventos necessário para o bom funcionamento do nosso sistema.

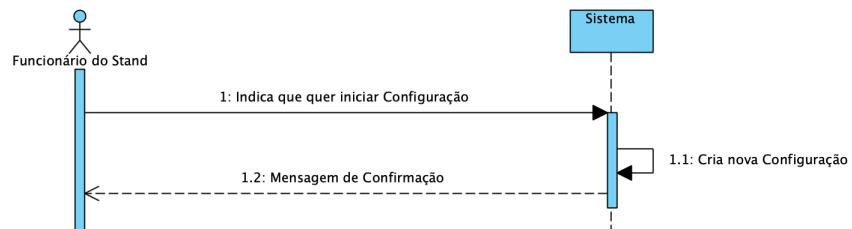
9.1 Configuração Ótima



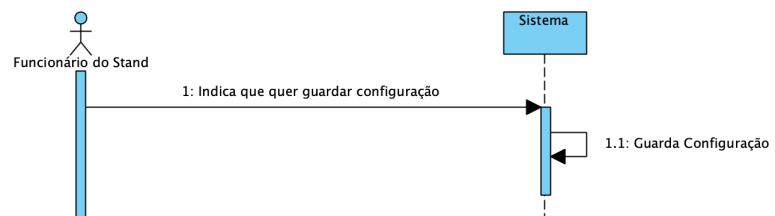
9.2 Criar Encomenda



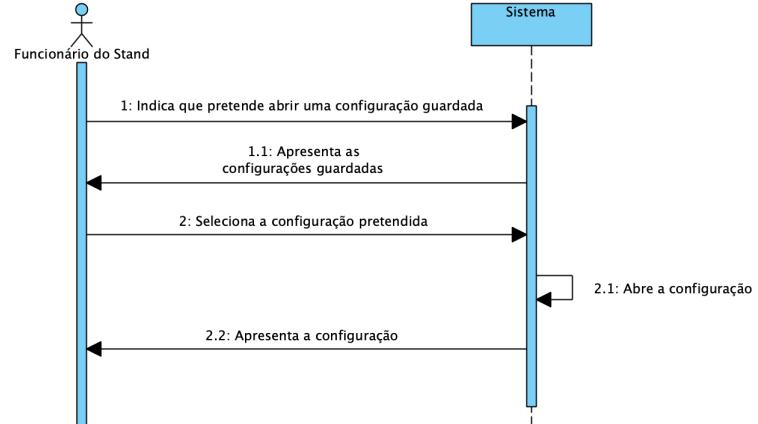
9.3 Iniciar Configuração



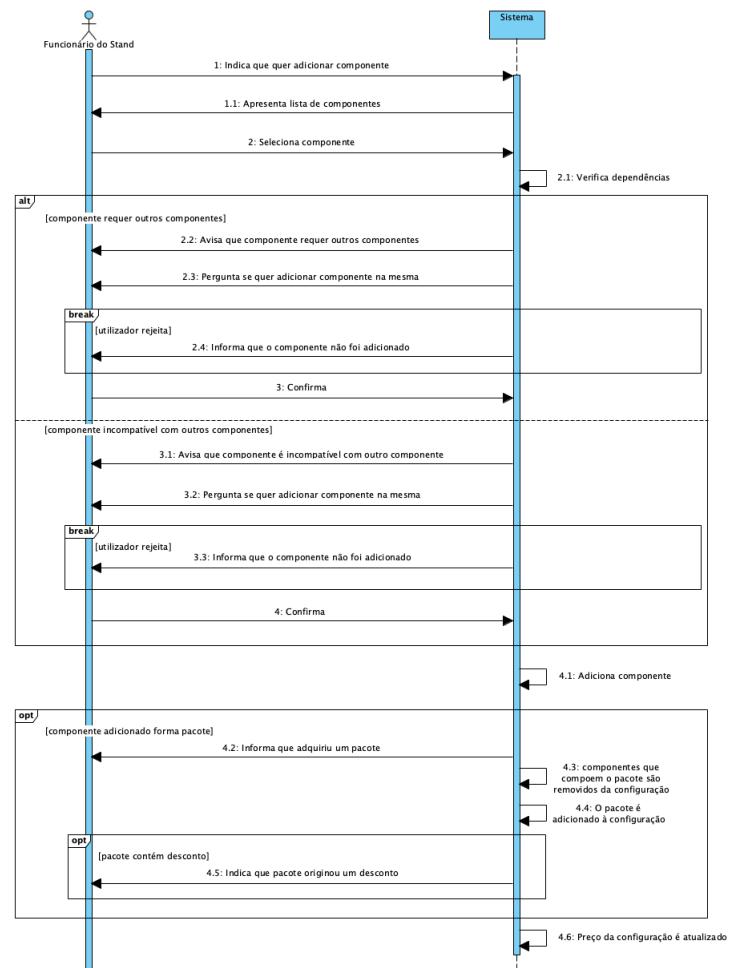
9.4 Guardar Configuração



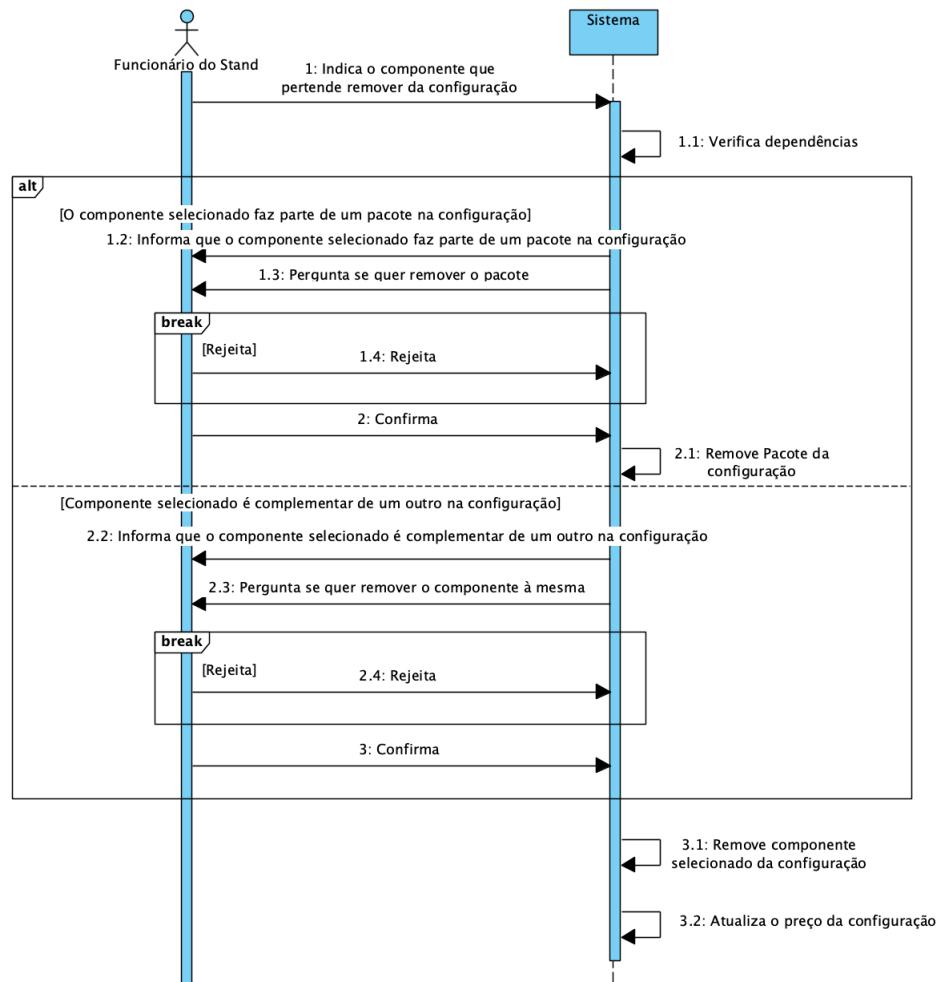
9.5 Abrir Configuração Guardada



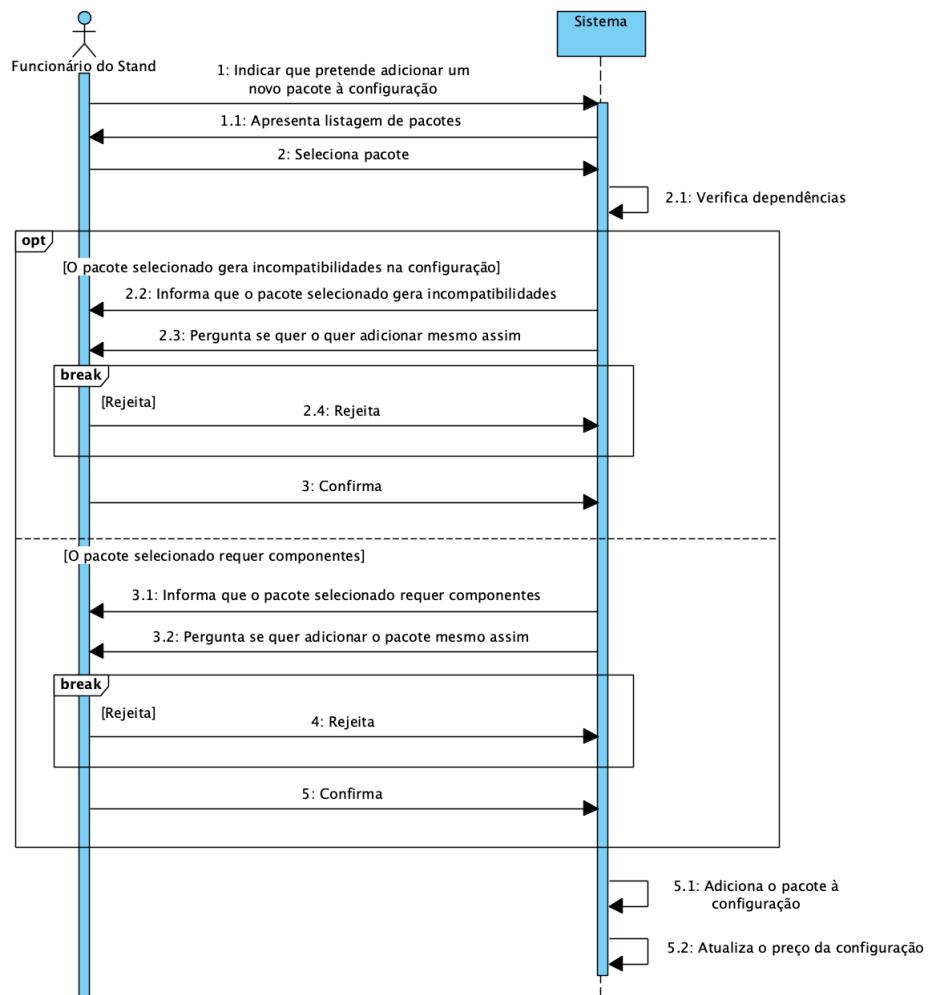
9.6 Adicionar Componente



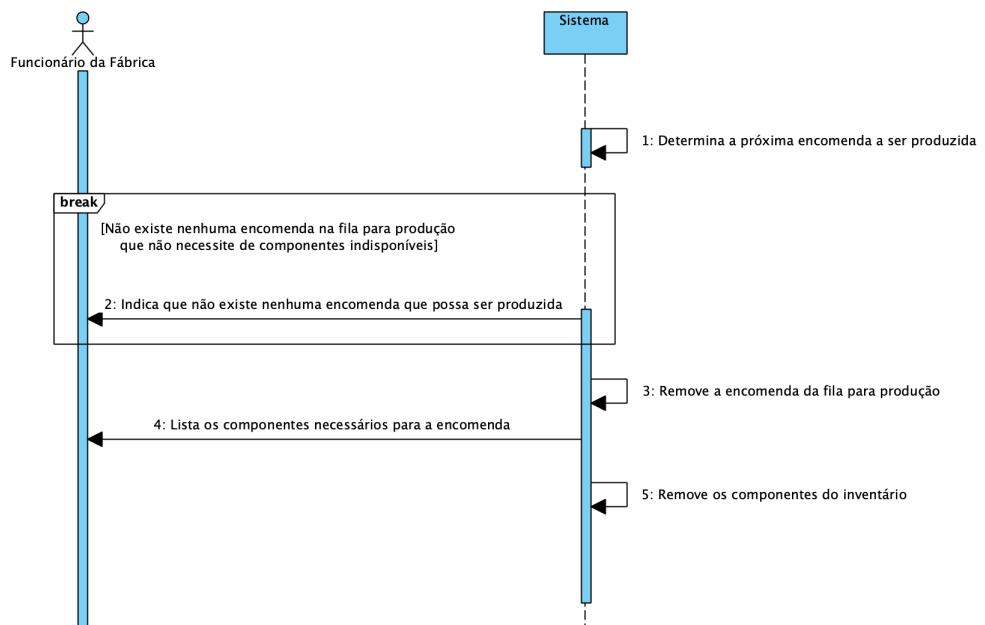
9.7 Remover Componente



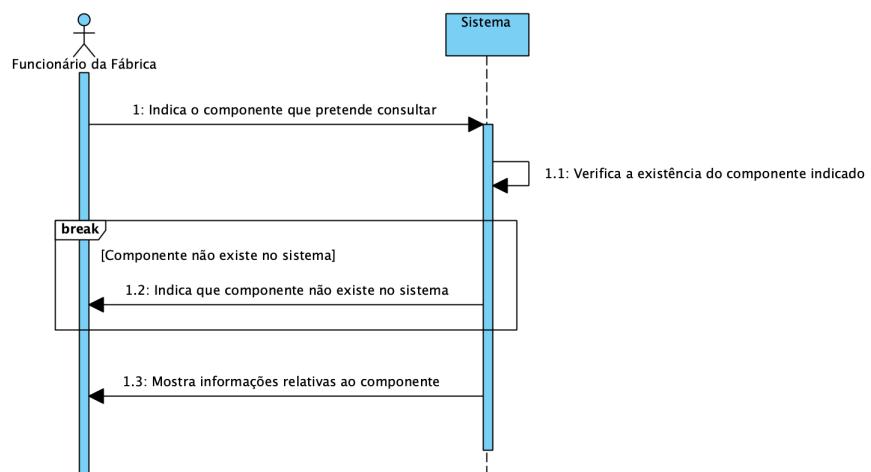
9.8 Adicionar Pacote



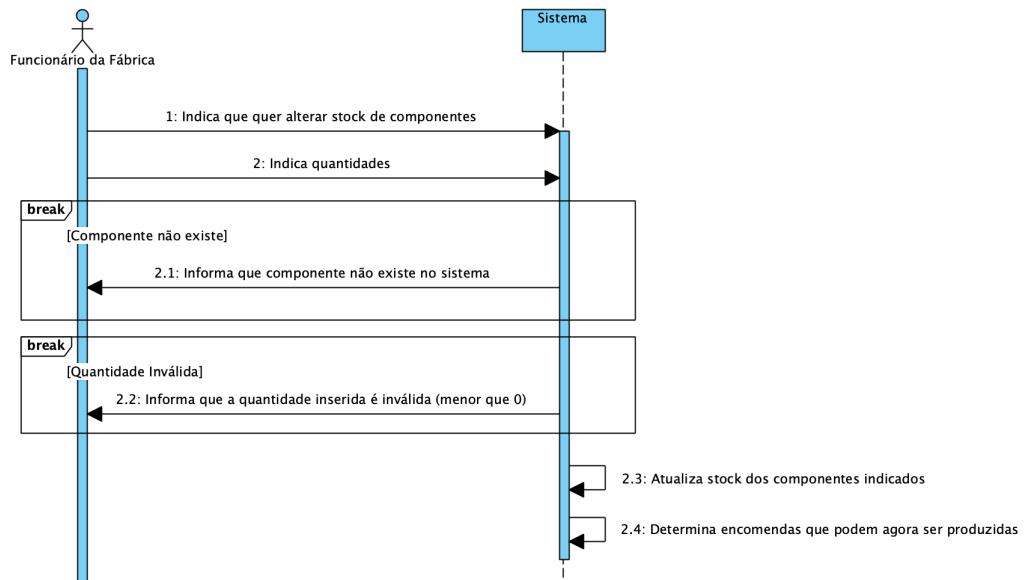
9.9 Consultar próxima encomenda com material



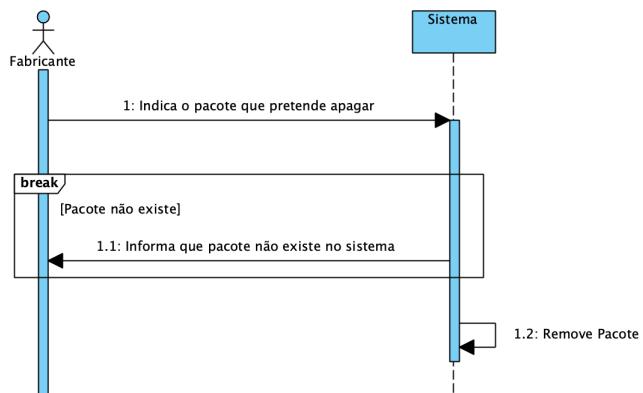
9.10 Consultar informação do componente



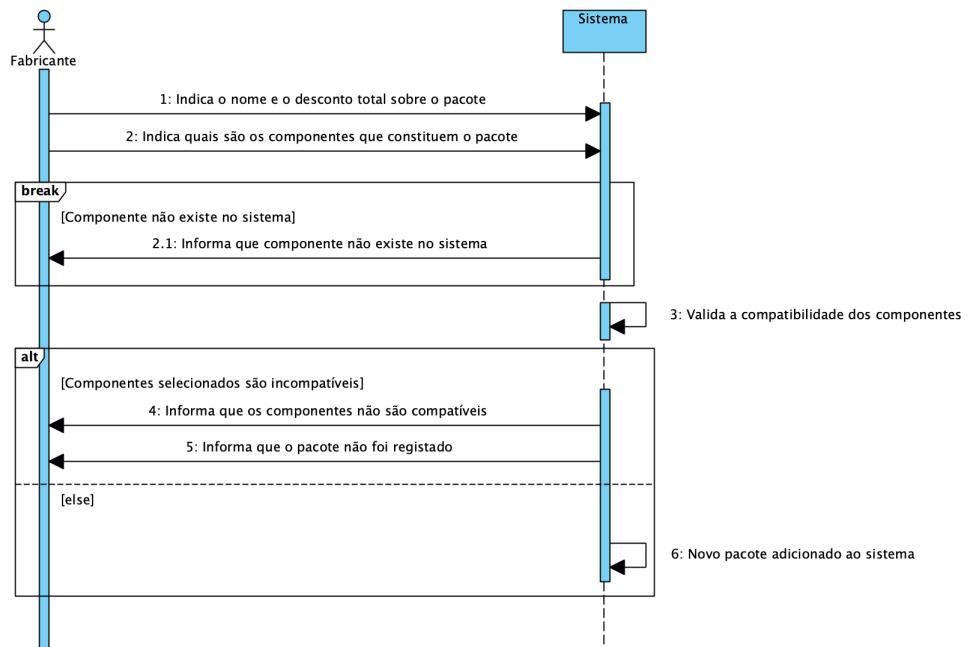
9.11 Atualizar Stock



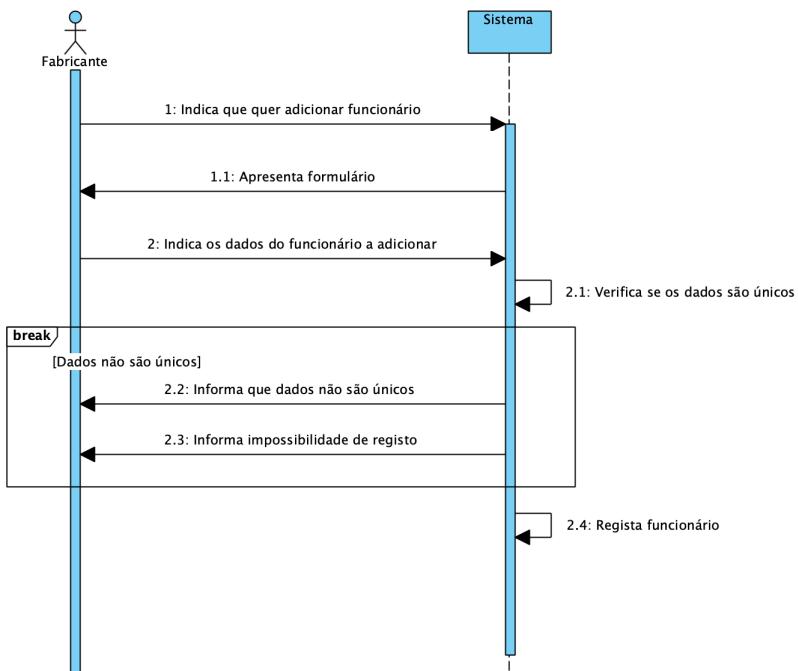
9.12 Apagar Pacotes



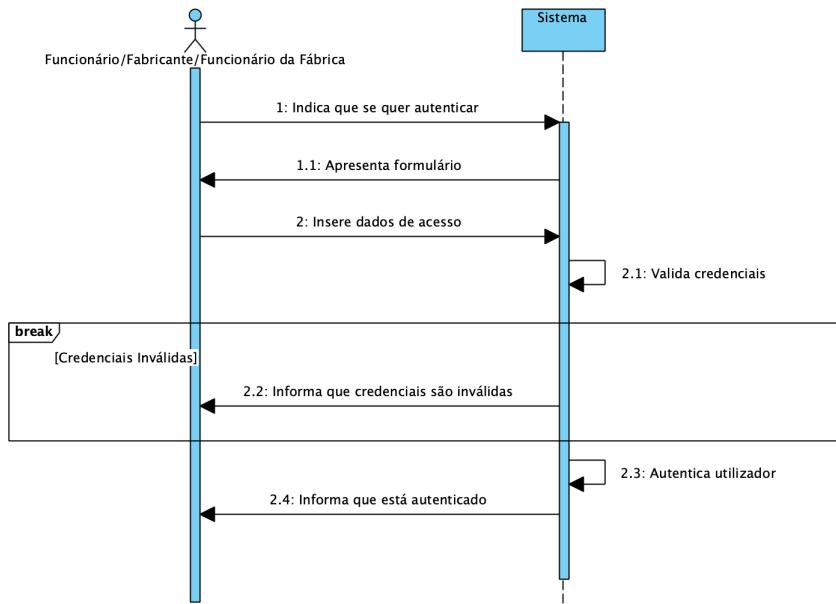
9.13 Adicionar Novo Pacote



9.14 Adicionar Funcionário

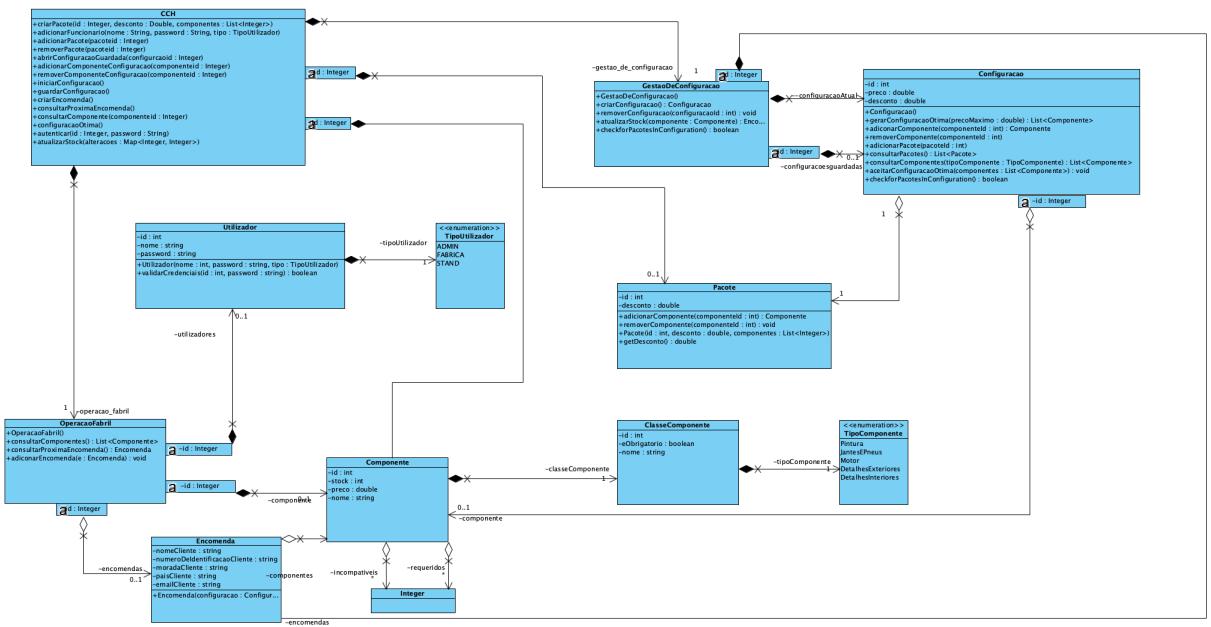


9.15 Autenticar



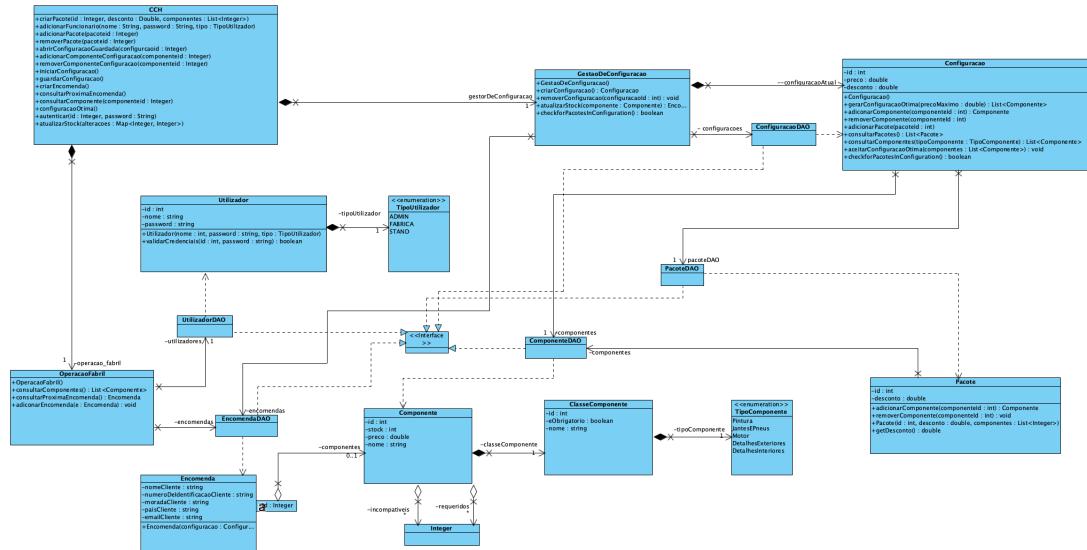
10 Diagrama de Classes

Com base no modelo de domínio foram selecionadas as componentes que dariam origem a classes, como por exemplo, a Configuração e o Utilizador. Acrescido a estas foram também adicionadas algumas novas classes que implementam a lógica de negócios, como por exemplo, a classe CCH ou a classe GestaoDeConfiguracao. Em seguida, para definir a interface, foram analisados todos os diagramas de sequência de sistema, identificadas todas as responsabilidades de cada classe e, então, foram modeladas com atributos e operações. Deste processo resultou o seguinte diagrama de classes.



11 Diagrama de ORM

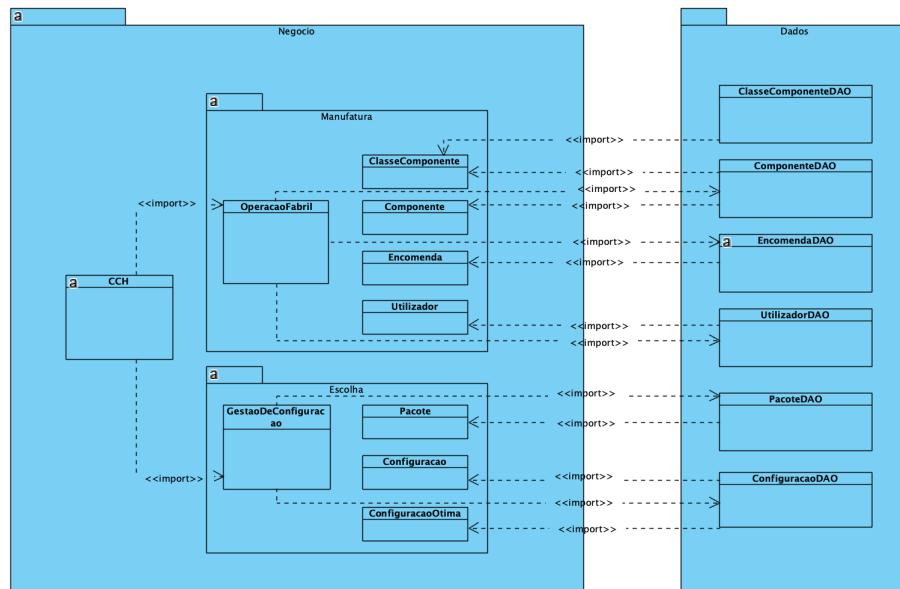
Após completo o diagrama de classes prosseguimos com o desenvolvimento do diagrama de Object Relational Mapping – ORM. Foram identificadas as entidades que têm estado, que têm necessidade de “armazenar” dados e, por isso, devem ser persistidas. Para além disso foram mapeados os relacionamentos, substituíram-se todas as coleções e maps por acessos à base de dados, recorrendo ao uso de DAO’s.



12 Diagrama de Packages

Na estruturação dos packages do nosso projeto, começamos por agrupar as classes correspondiam à mesma camada na arquitetura 3-tier. Desta forma, torna-se evidente a capacidade da nossa solução de software modular e flexível de lidar com futuras alterações nos requisitos tecnológicos e incorporação em outras aplicações.

Posteriormente, separamos a camada de negócios em Manufatura e em Escolha para diferenciar os componentes da aplicação que correspondem às entidades do stand e da fábrica.

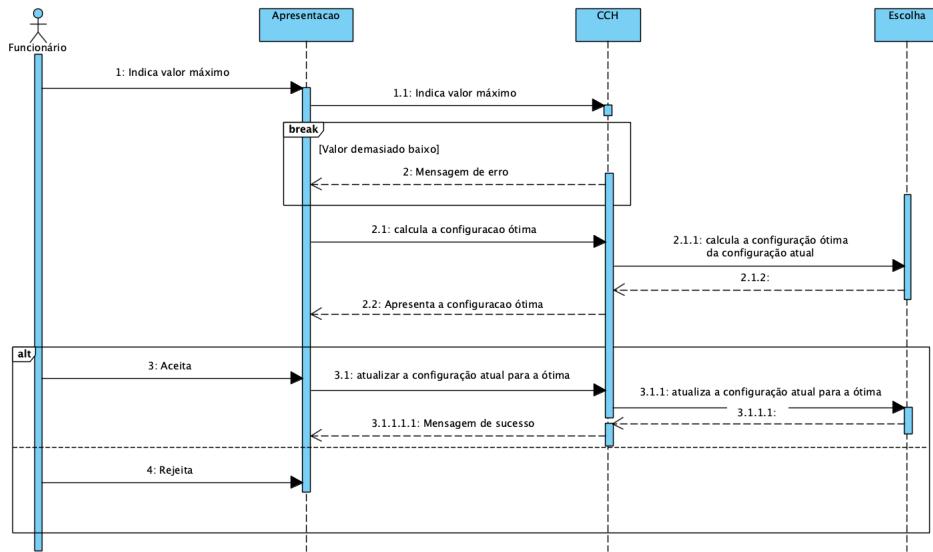


13 Diagrama de Sequência de Subsistemas

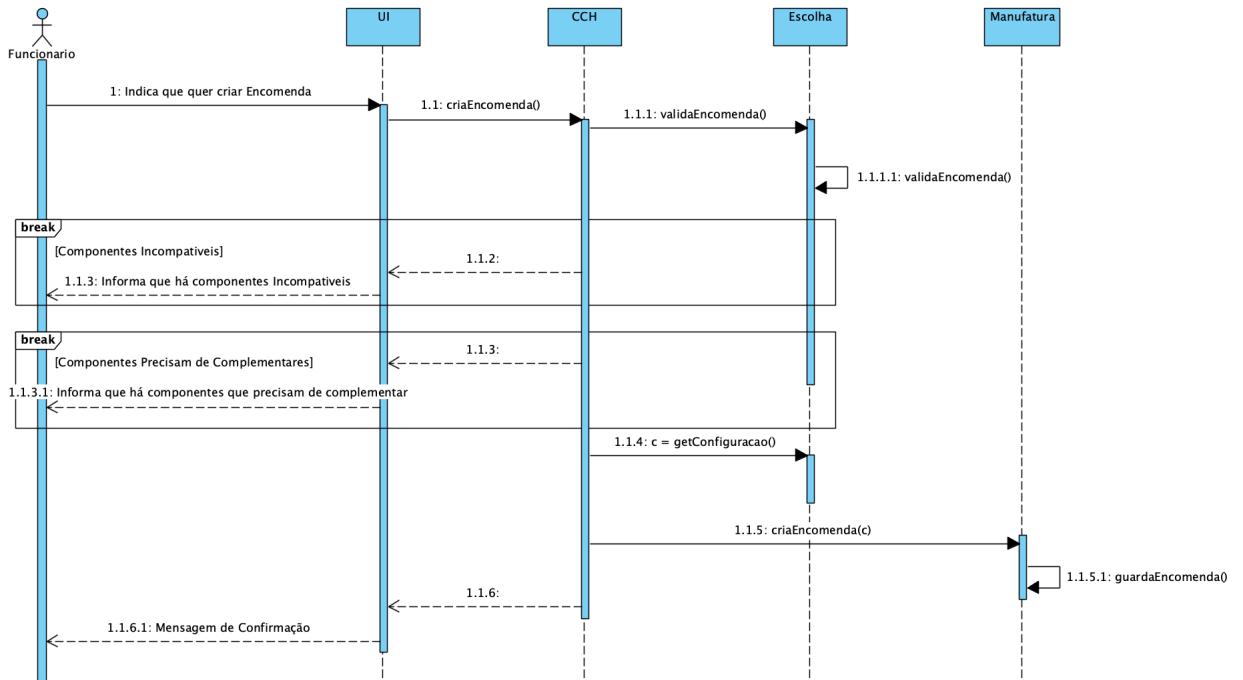
Através da modelação do Diagrama de Packages identificamos dois subsistemas: Manufatura e Escolha. Para além destes dois subsistemas é garantido que o sistema terá a Interface Gráfica e o Facade (CCH).

Desta forma, podemos voltar a analisar cada um dos use cases identificados e procedermos a um estudo mais detalhado dos mesmos, adicionando-lhes as novas informações. Para isso, recorremos aos Diagramas de Sequência de Subsistemas, onde desdoblarmos os Diagramas de Sequência de Sistemas presentes na Secção 9.

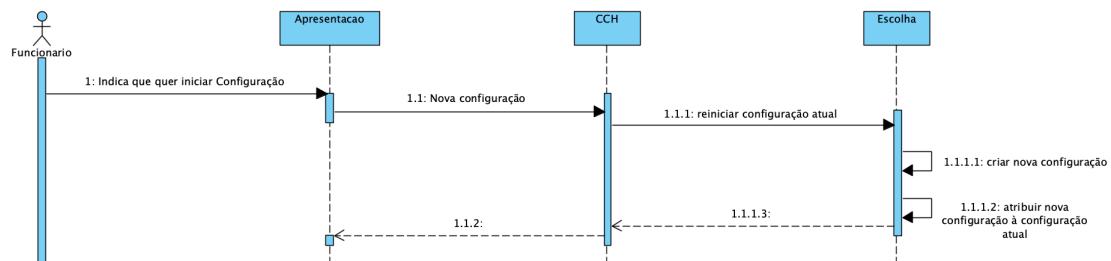
13.1 Configuração Ótima



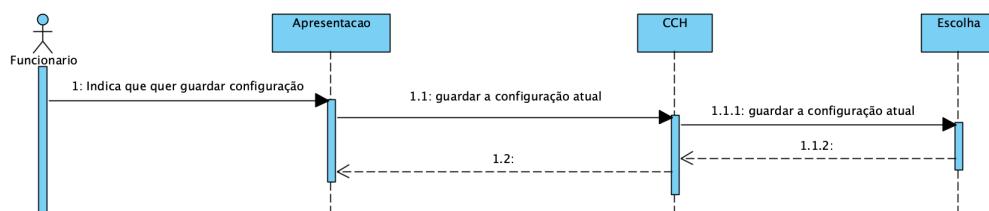
13.2 Criar Encomenda



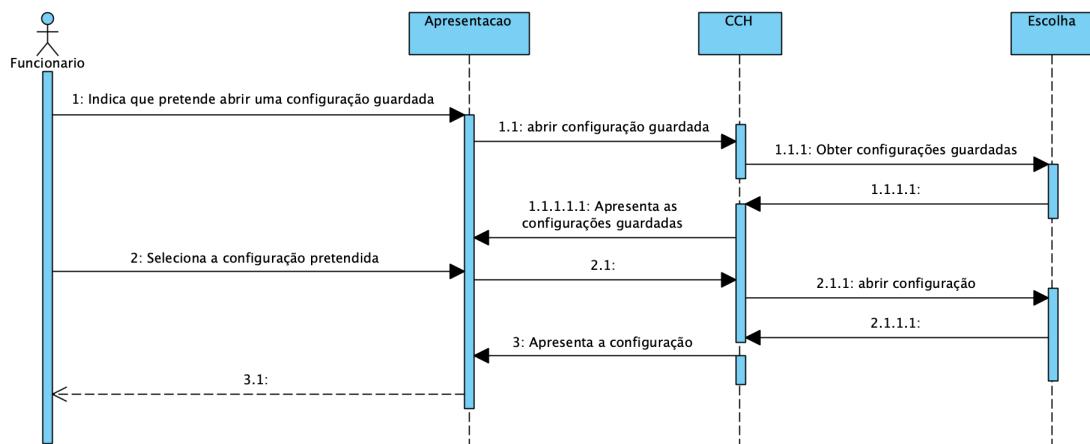
13.3 Iniciar Configuração



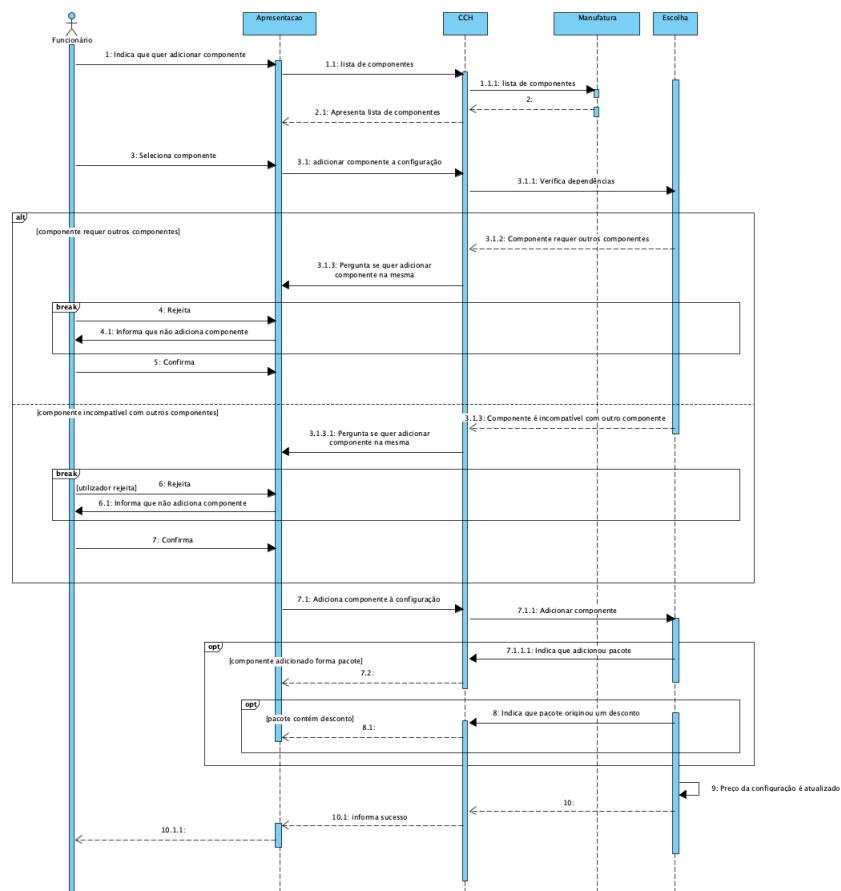
13.4 Guardar Configuração



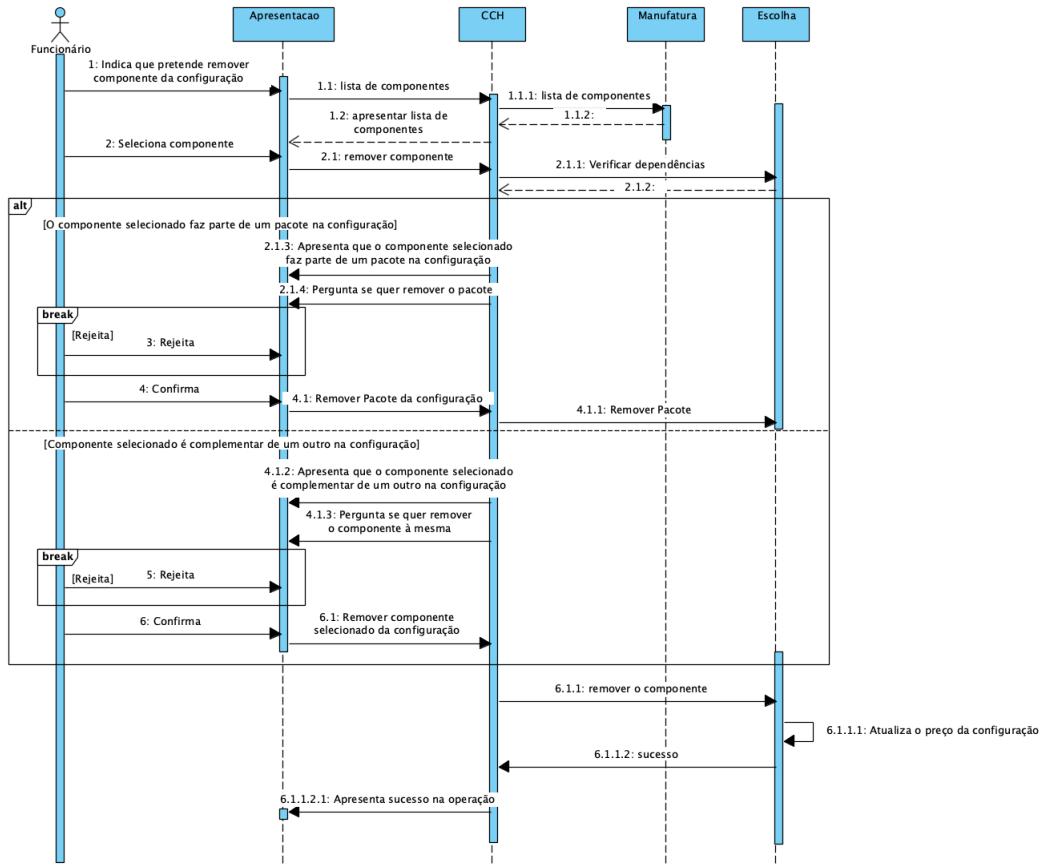
13.5 Abrir Configuração Guardada



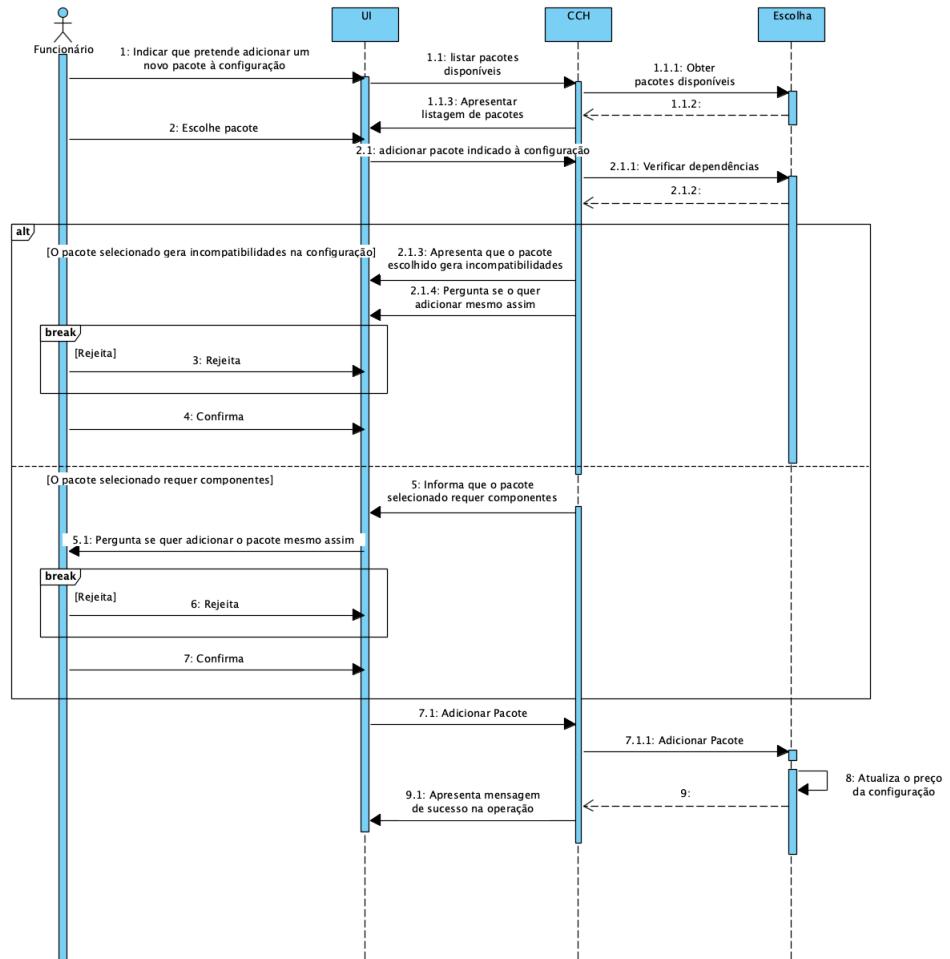
13.6 Adicionar Componente



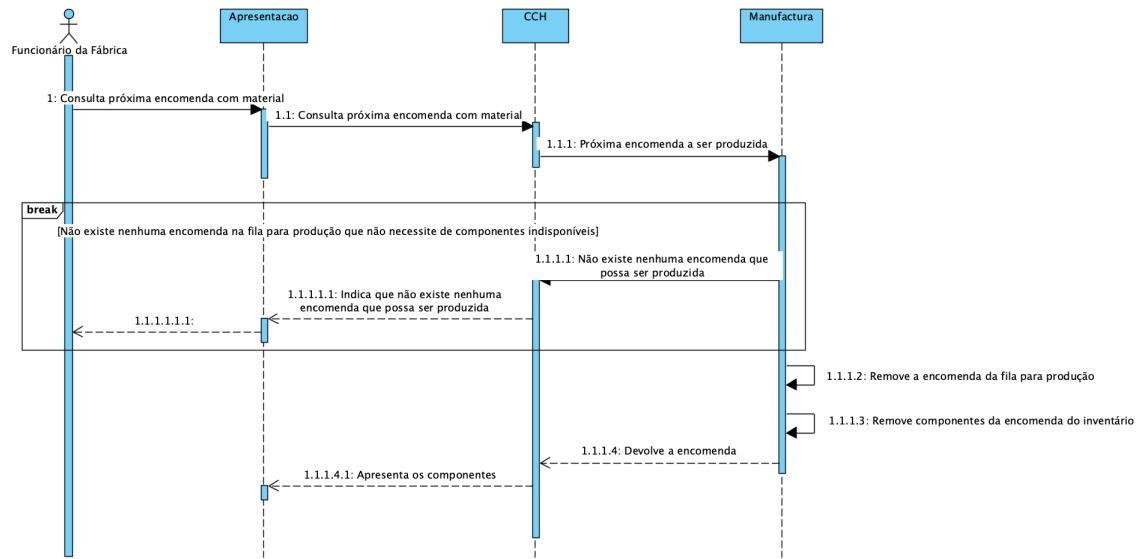
13.7 Remover Componente



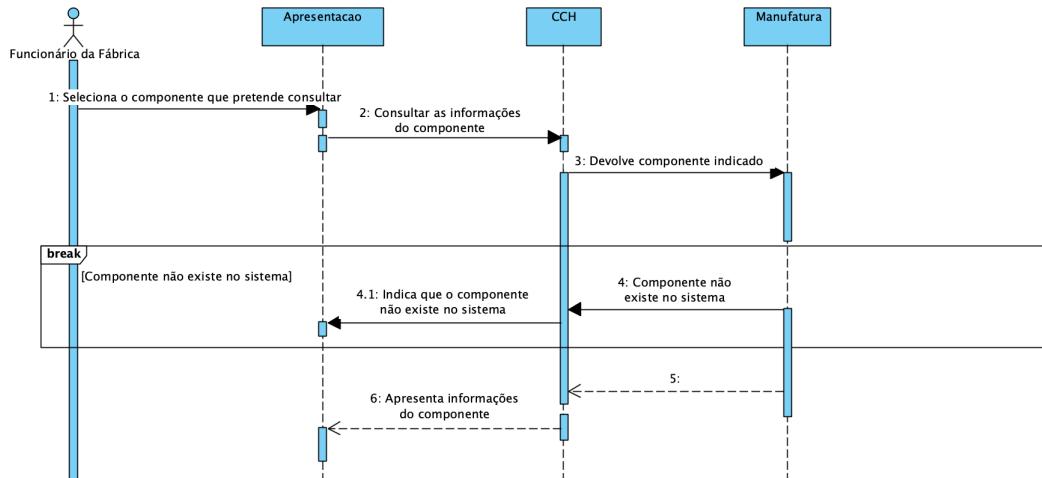
13.8 Adicionar Pacote



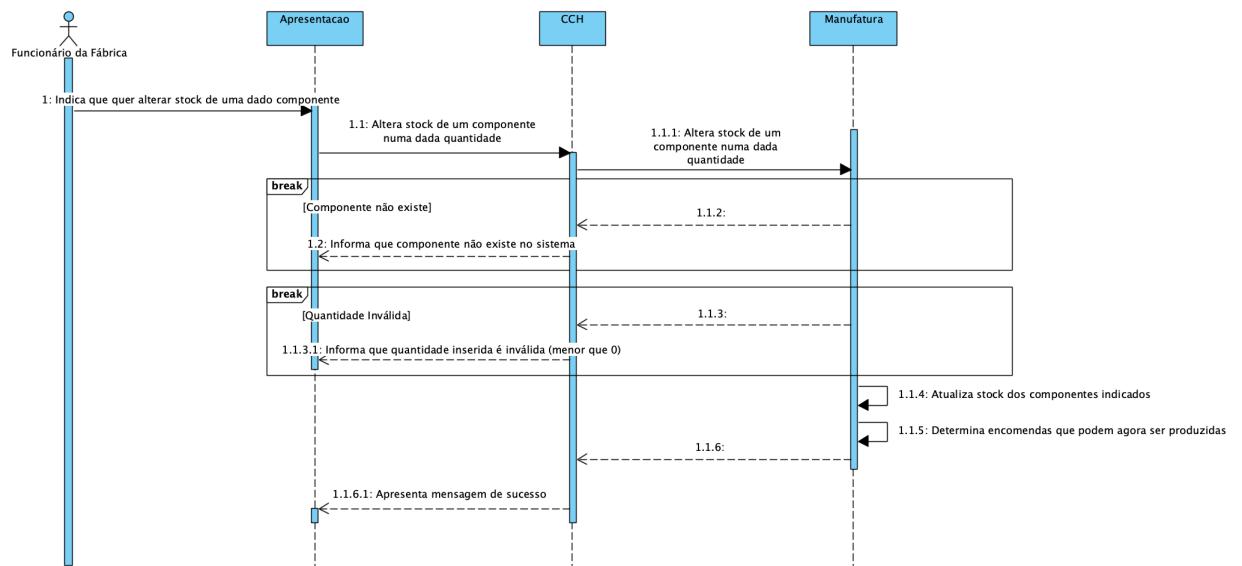
13.9 Consultar próxima encomenda com material



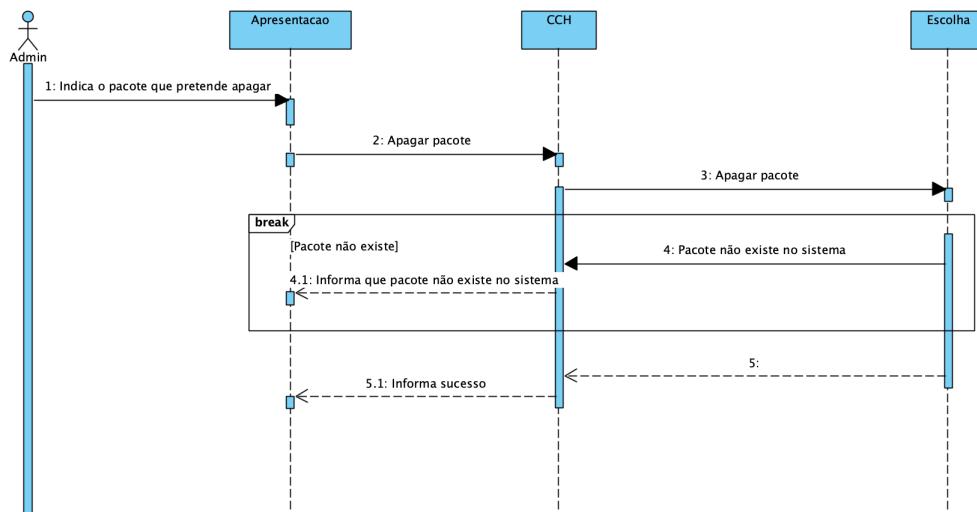
13.10 Consultar informação do componente



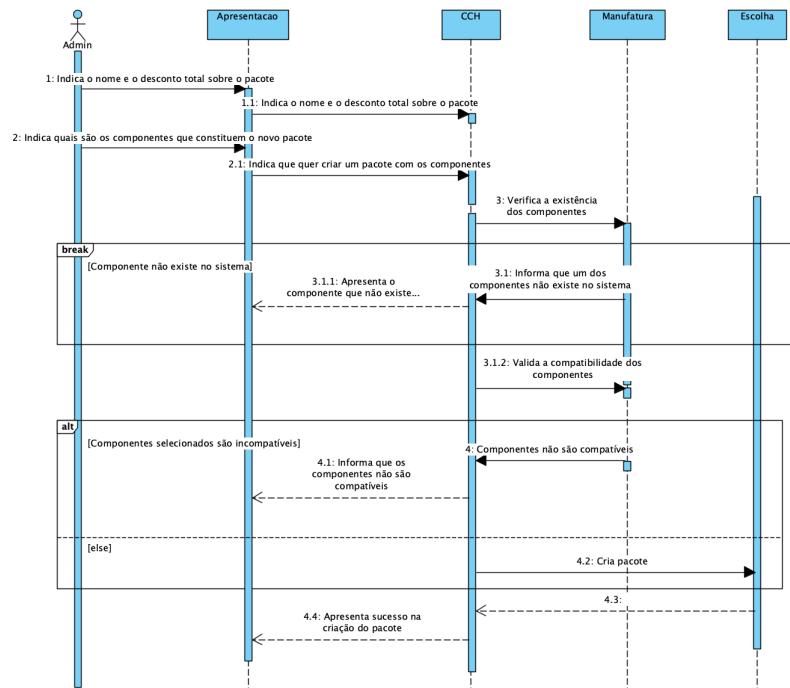
13.11 Atualizar Stock



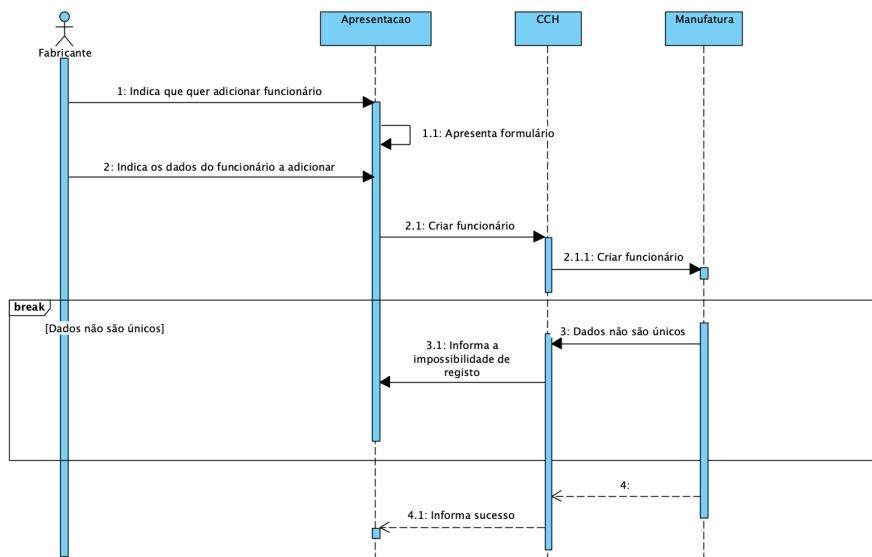
13.12 Apagar Pacotes



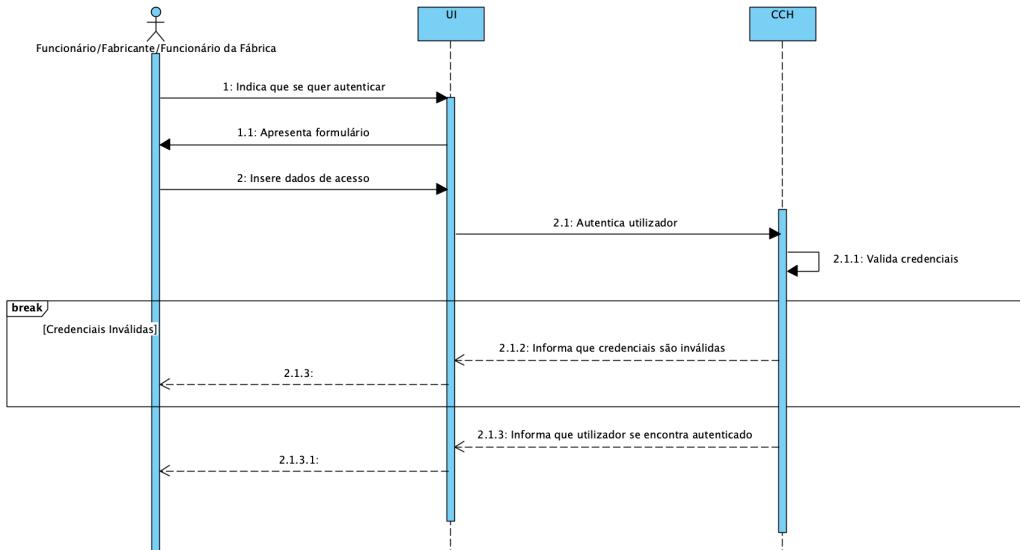
13.13 Adicionar Novo Pacote



13.14 Adicionar Funcionário



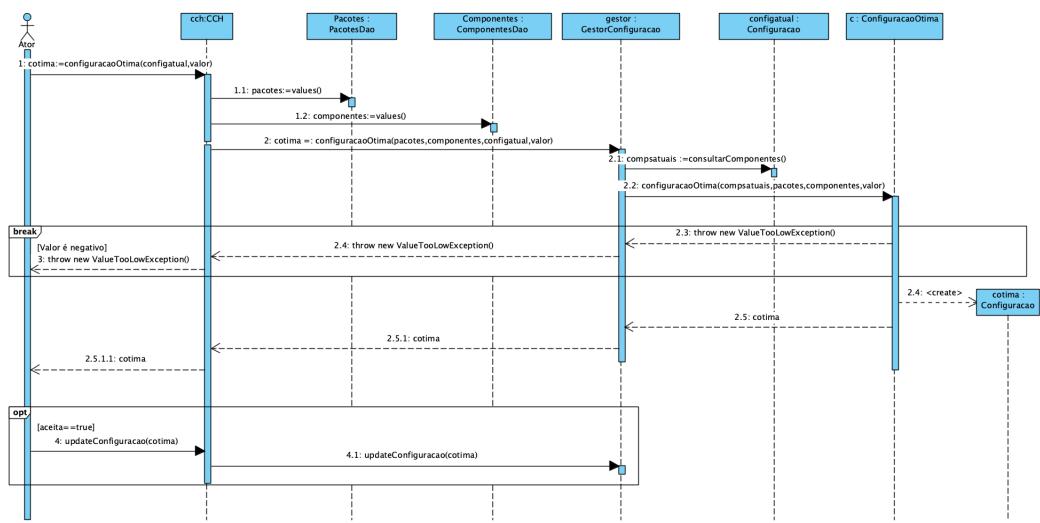
13.15 Autenticar



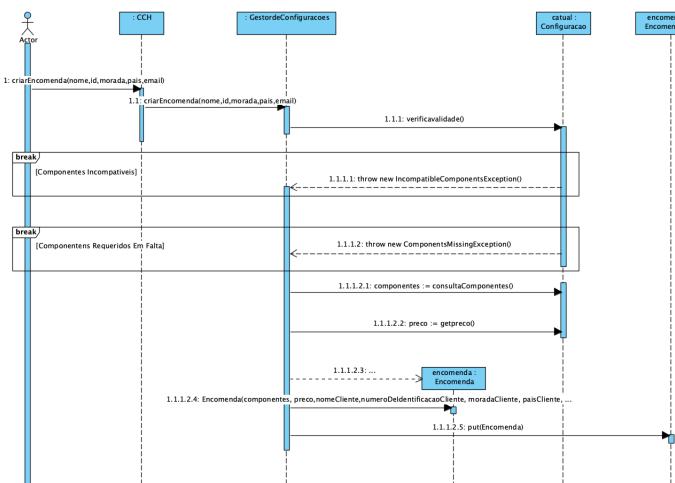
14 Diagrama de Sequência de Implementação

Na sequência dos diagramas de sequência de subsistemas foram desenvolvidos os diagrama de sequência de implementação, que se distinguem por ser um refinação dos anteriores. Nesta fase já se faz referência a classes e às suas instâncias específicas, é também já utilizada uma sintaxe muito próxima da que será usada no código Java. Isto permite começar o processo de transição entre os diagramas e a implementação do projeto. Estruturalmente tudo está já muito bem definido e, após a geração automática de código ficará apenas a faltar desenvolver os corpos dos métodos. Logicamente, nesta fase, as responsabilidades já estão bem distribuídas entre as várias classes e está garantida a modularização do código.

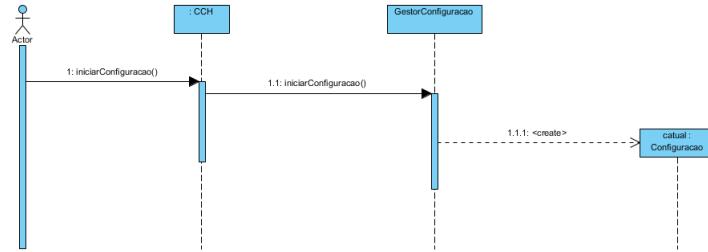
14.1 Configuração Ótima



14.2 Criar Encomenda



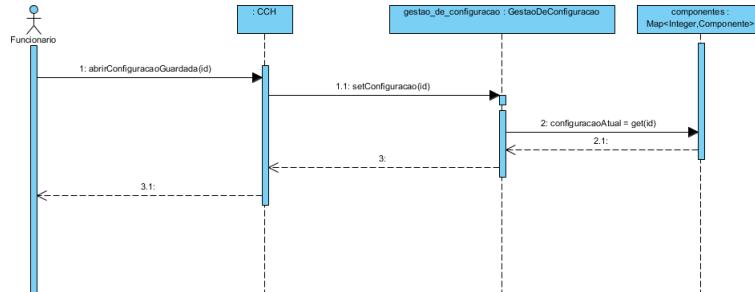
14.3 Iniciar Configuração



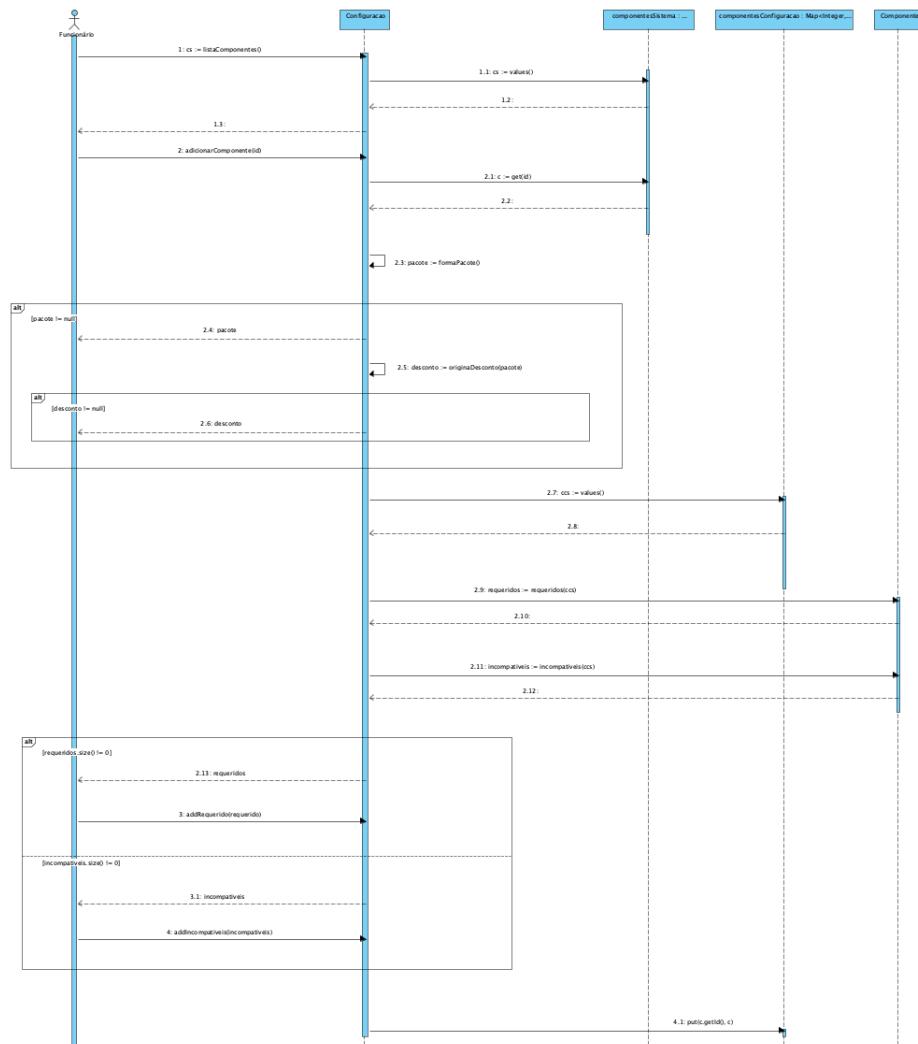
14.4 Guardar Configuração

Este use case foi abandonado na fase de implementação devido à introdução dos ORM. A introdução destes fez com que as configurações passassem a ser automaticamente guardadas na base de dados.

14.5 Abrir Configuração Guardada



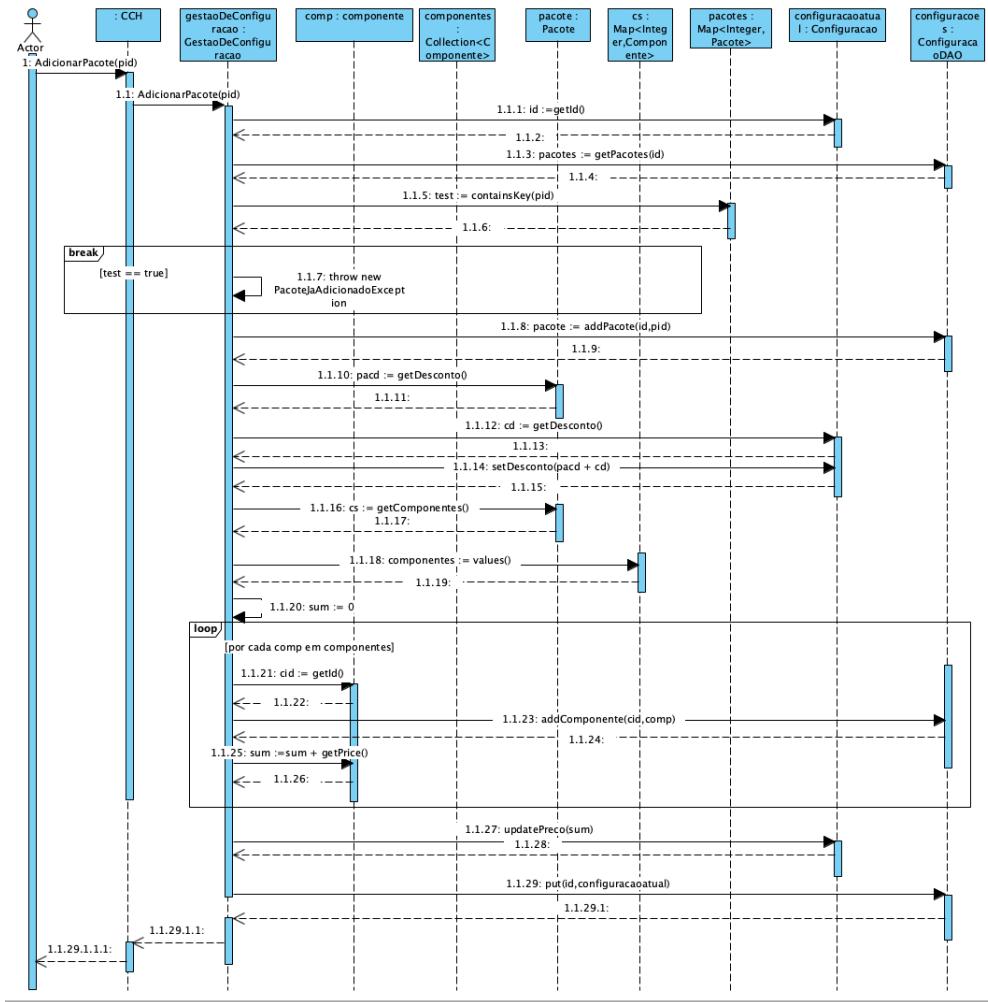
14.6 Adicionar Componente



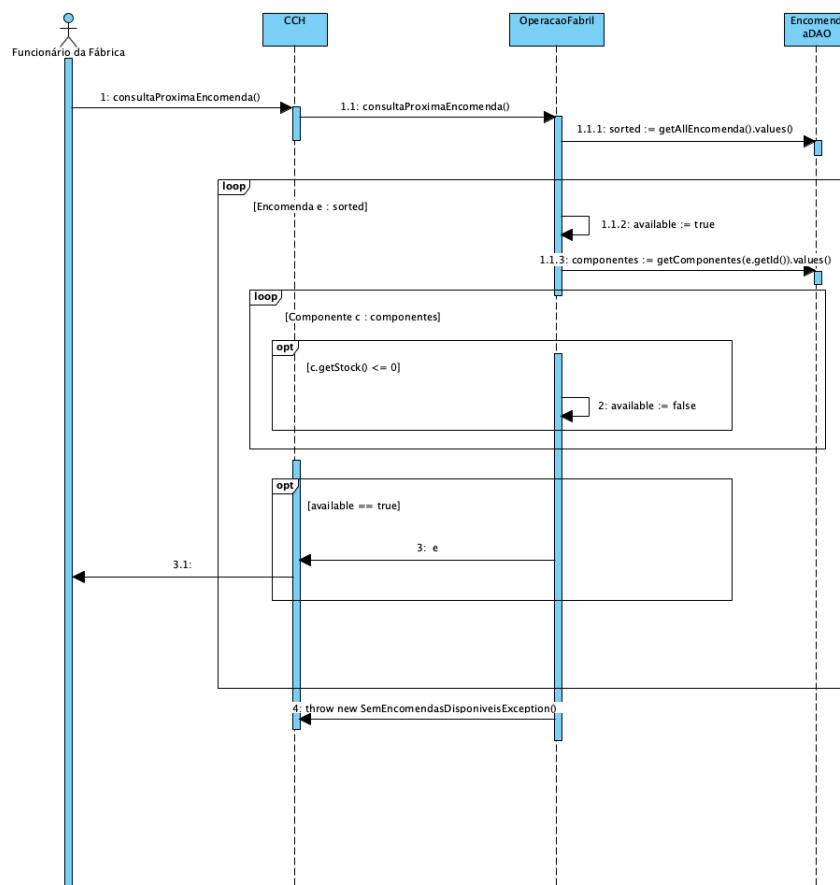
14.7 Remover Componente

O seguinte diagrama não cabia no relatório portanto vai em anexo.

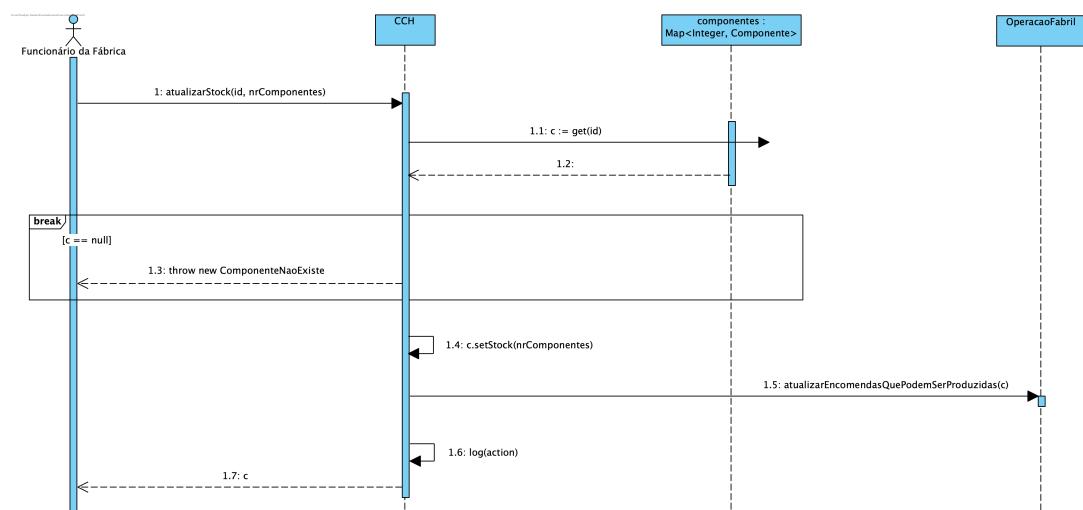
14.8 Adicionar Pacote



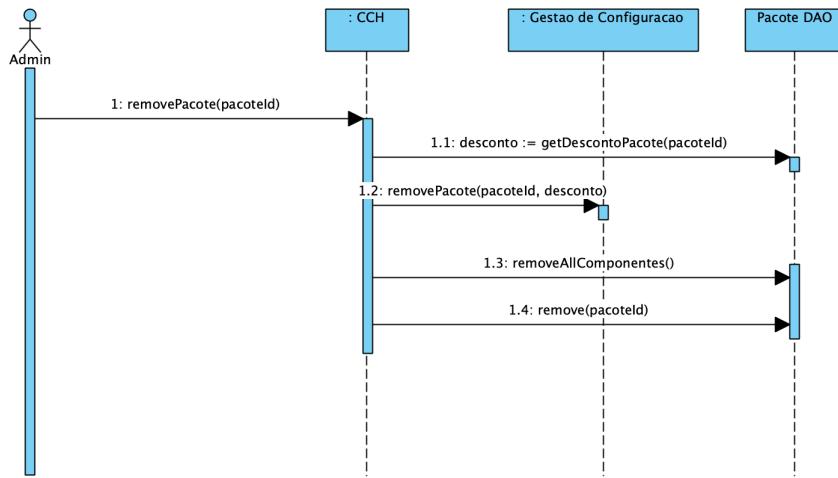
14.9 Consultar próxima encomenda com material



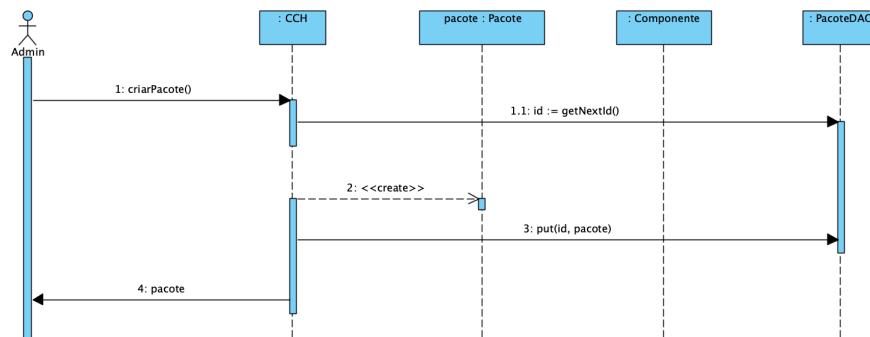
14.10 Atualizar Stock



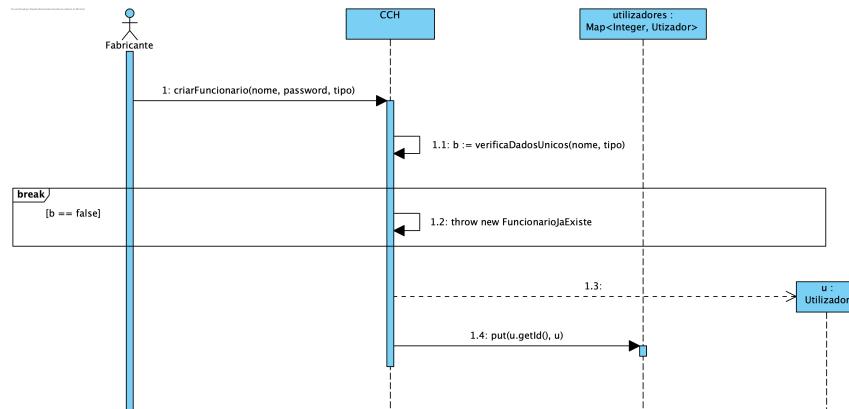
14.11 Apagar Pacotes



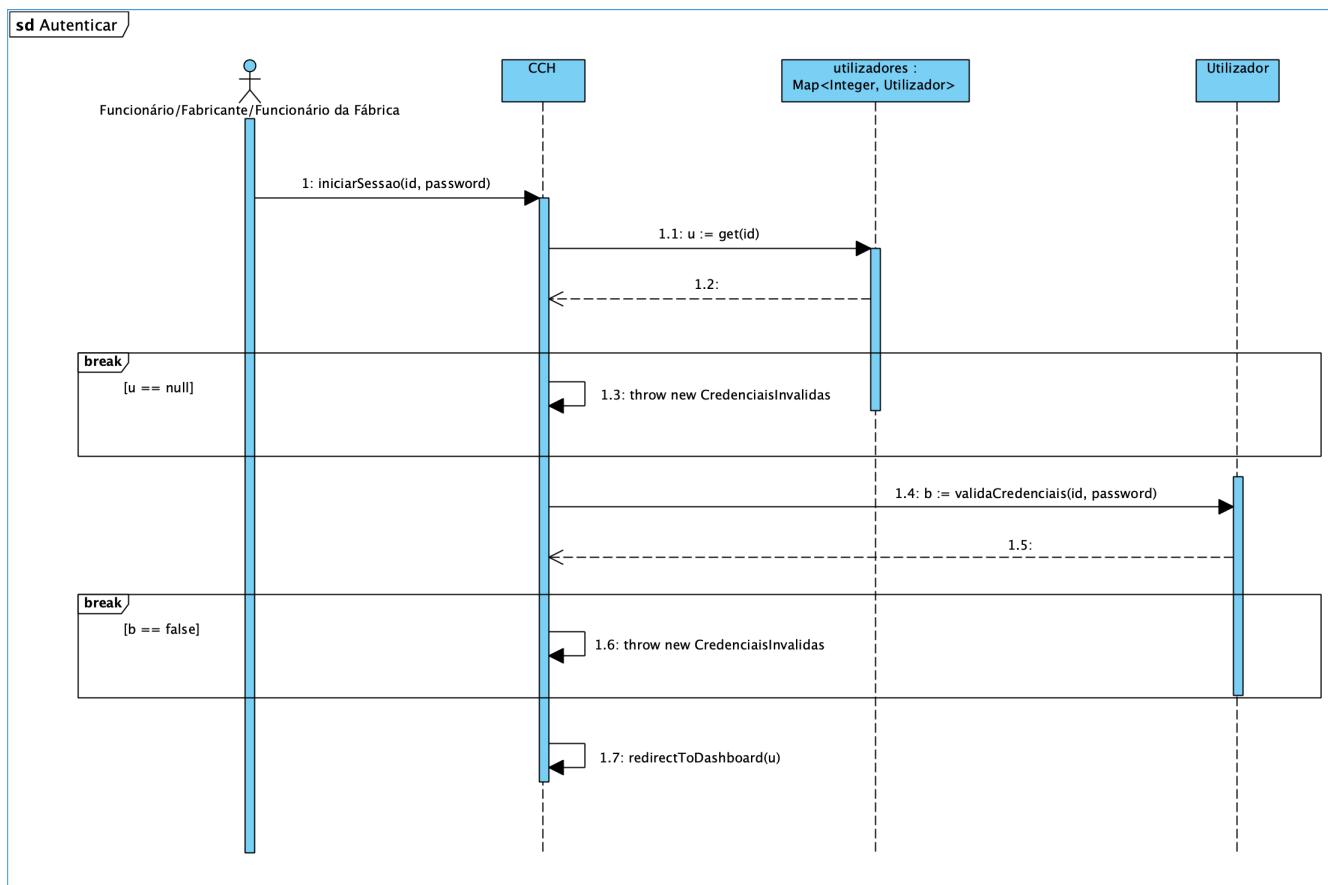
14.12 Adicionar Novo Pacote



14.13 Adicionar Funcionário



14.14 Autenticar



15 Base de Dados

De modo a persistirmos a nossa aplicação e, consequentemente, deixarmos de guardar os dados em memória, construímos uma base de dados que servirá para armazenar todas as informações necessárias para o funcionamento do nosso software.

O sistema de gestão de base de dados relacional escolhido para implementar a base de dados foi o MySQL e em anexo encontra-se o script de criação da base de dados e um script de povoamento, meramente exemplificativo, para a aplicação poder ser testada.

O modelo lógico da base de dados pode ser visualizado na figura seguinte.

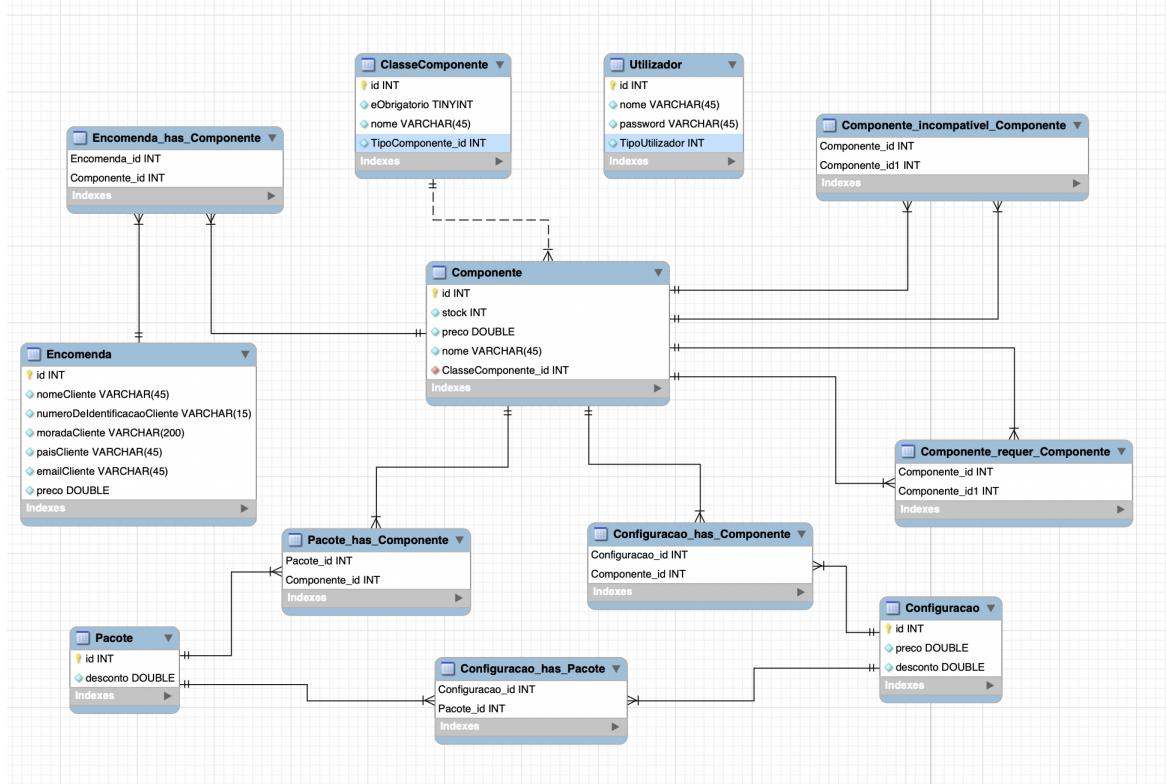


Figura 54: Modelo Lógico da Base De Dados.

16 Modelo de Programação Inteira para a Configuração Ótima

De forma a resolvemos o problema da Configuração Ótima modulamos o problema com programação inteira e recorremos ao solver CPLEX da IBM para resolver o modelo criado.

O modelo utilizado foi o seguinte:

- P_x - variável inteira booleana que toma o valor 1 se o pacote x foi selecionado na configuração e 0 se não.
- C_x - variável inteira booleana que toma o valor 1 se o componente x foi selecionado na configuração e 0 se não.

O modelo tem como função objetivo a maximização da soma dos preços dos componentes utilizados.

- Função Objetivo:

$$\text{Max} \sum \text{PrecoComponente}_x * C_x \quad (1)$$

De forma a impedir que o modelo não selecionasse todos os componentes foram definidas restrições para os seguintes casos:

O custo da configuração não pode ser superior ao valor fornecido logo é adicionada a restrição:

- $\sum \text{PrecoComponente}_x * C_x - \sum \text{Desconto}_x * P_x \leq \text{Valor}$ (2)

Não podem ser selecionados pacotes que partilhem componentes.

- Exemplo: O componente 1 está nos pacotes 1 e 4 então é adicionada a restrição:

$$P1 + P4 + P1 \leq 1 \quad (3)$$

A seleção de um pacote implica a seleção dos seus componentes

- Exemplo: O pacote 5 tem os componentes 3 e 4 então são adicionadas as restrições:

$$P5 - C3 \leq 0 \text{ e } P5 - C4 \leq 0 \quad (4)$$

Se um componente requerer outros componentes esses devem ser selecionados

- Exemplo: O componente 2 requer os componentes 1 e 5 então são adicionadas as seguintes restrições:

$$C2 - C1 \leq 0 \text{ e } C2 - C5 \leq 0 \quad (5)$$

Não podem ser selecionados componentes incompatíveis em simultâneo

- Exemplo: O componente 1 é incompatível com o componente 9 e com o 10 então são adicionadas as seguintes restrições:

$$C1 + C9 \leq 1 \text{ e } C1 + C10 \leq 1 \quad (6)$$

A configuração deve conter os componentes que já foram escolhidos pelo utilizador.

- Exemplo: O utilizador selecionar os componentes 2, 4 e 6 então são adicionadas as seguintes restrições:

$$C2 = 1, C4 = 1 \text{ e } C6 = 1 \quad (7)$$

17 Implementação

Em termos da implementação do projeto, o primeiro passo foi a geração automática de código através dos vários diagramas que tinham sido desenvolvidos anteriormente. Neste momento estavam já definidos estruturalmente todo os métodos necessários e faltava apenas implementá-los.

O segundo passo foi então a já referida implementação deste métodos. Sempre acompanhados dos diagramas de sequência de implementação, este processo foi relativamente fácil porque bastou apenas seguir a estrutura já modelada e transformá-la em código Java, o que necessitou apenas da adição de alguma lógica adicional, nomeadamente na manipulação de dados e controlo de fluxo.

A parte mais trabalhosa da implementação foi, sem dúvida, a construção da parte gráfica. Apesar de toda a parte visual ter sido apenas um adaptar da prototipagem da UI para JavaFX, toda a lógica de controladores foi um processo algo custoso e que causou alguns problemas. Ainda assim, quando começamos a dominar a ferramenta JavaFX tudo ficou bastante simplificado. Toda a ligação com entre a interface e a lógica de negócio estava muito facilitada por existir uma API bastante fácil de usar e que fornecia todas as operações necessárias.

17.1 Implementação Configuração Ótima

A configuração ótima foi implementada numa classe à parte de forma a facilitar o mais possível a troca da solução por outra, caso fosse necessário alterar o solver utilizado ou calcular a configuração fora do sistema por exemplo.

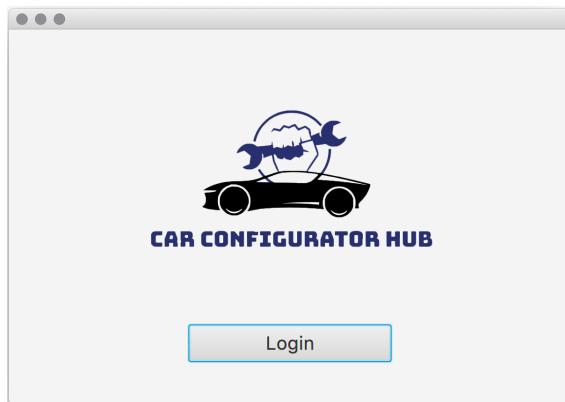
A implementação do algoritmo passou por a geração das restrições do modelo de programação linear no código, tendo em conta os componentes e os pacotes que se encontram no sistema.

A medida que estas restrições são geradas estas são alimentadas ao solver CPLEX.

Depois de geradas as restrições chama-se o método do CPLEX que resolve o modelo. Se o CPLEX achar a solução do modelo gerado convertem-se as variáveis de decisão nos componentes e pacotes que foram selecionados e adiciona-se os componentes à configuração ótima e retorna-se a solução. Caso o modelo não tenha solução é atirada uma exceção.

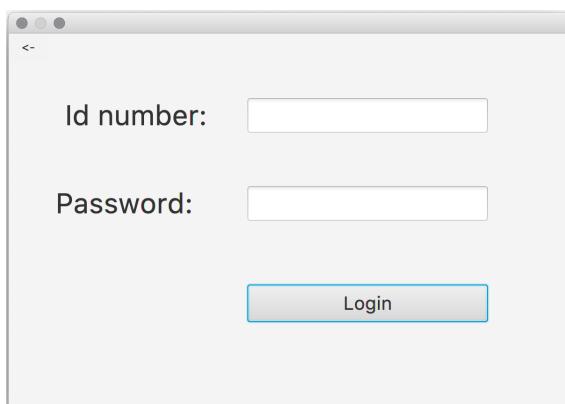
18 Manual de Utilização

Após o utilizador comum do software iniciar a aplicação é confrontado com a janela representada na figura seguinte, onde pode ver o logótipo da Car Configurator Hub e tem a possibilidade de abrir o ecrã onde poderá efetuar o login na aplicação.

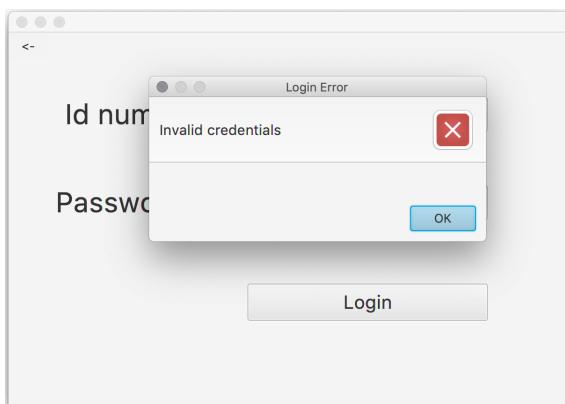


Botão: Login

Carregando na opção de “Login”, é apresentada a seguinte janela:



Caso não exista nenhum utilizador com o id indicado ou a password estiver incorreta irá aparecer uma mensagem no ecrã informando que as credenciais inseridas são inválidas.



Em todos os ecrãs o utilizador tem a possibilidade de regressar ao ecrã anterior, clicando na

seta do canto superior esquerdo.

A aplicação, a par do sistema, está dividida em três tipos de utilizadores:

- Administrador;
- Funcionário do Stand;
- Funcionário da Fábrica.

Deste modo, após o utilizador inserir as suas credenciais e as mesmas estarem corretas, a aplicação reencaminha-lo-á para a respetiva janela tendo em conta a sua função na empresa.

Administrador

No caso das credenciais inseridas corresponderem a um administrador, o ecrã que apresentado será o seguinte:

Pacote	Componentes	Desconto (€)	
Pacote 0	Abrir	13.0	Apagar
Pacote 1	Abrir	1.0	Apagar
Pacote 2	Abrir	21.0	Apagar

Adicionar Pacote

Gerir Funcionários

Nesta janela o administrador consulta os pacotes que existem no sistema, e pode visualizar os componentes de cada um, editar o desconto do pacote, apagar o pacote e adicionar um novo pacote. Tem ainda um botão que o redirecionará para um ecrã que o permite fazer a gestão dos funcionários.

Botão: Adicionar Pacote

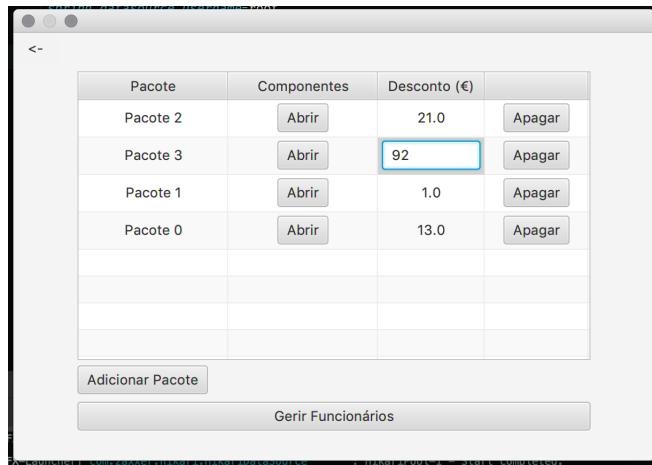
Clicando no botão de adicionar pacote será adicionado um novo pacote ao sistema com as informações default, ou seja, sem componentes e com desconto igual a 0.

Pacote	Componentes	Desconto (€)	
Pacote 1	Abrir	1.0	Apagar
Pacote 0	Abrir	13.0	Apagar
Pacote 3	Abrir	0.0	Apagar
Pacote 2	Abrir	21.0	Apagar

Adicionar Pacote

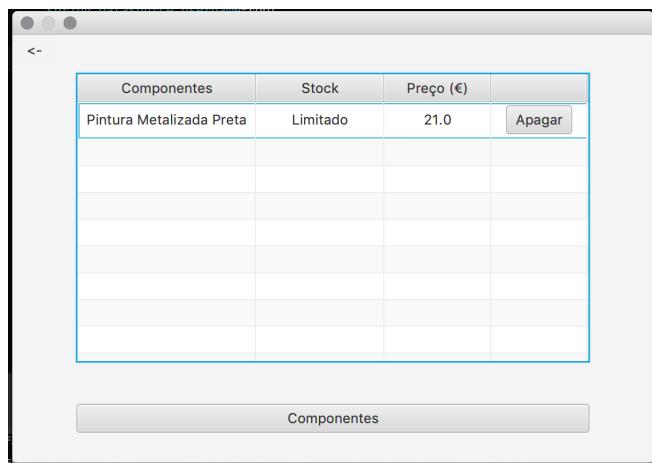
Gerir Funcionários

O administrador poderá alterar o desconto do pacote clicando duas vezes no valor do desconto:



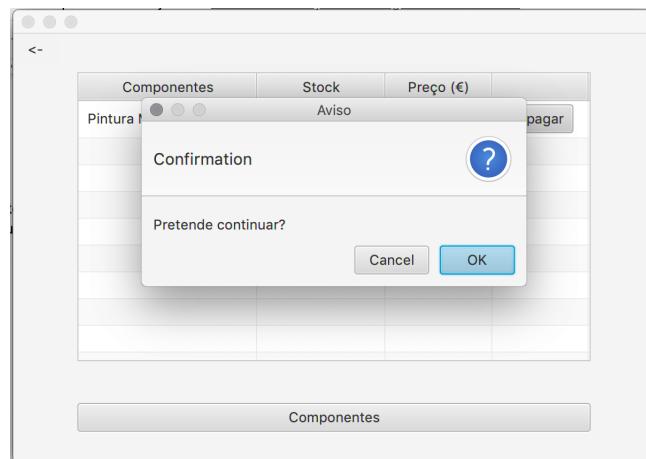
Botão: Apagar

Para apagar um determinado pacote o administrador deverá clicar no botão apagar. Irá aparecer-lhe uma mensagem para confirmar se deseja efetivamente apagar o pacote e, em caso afirmativo, o pacote é apagado do sistema (com as devidas implicações nas configurações).

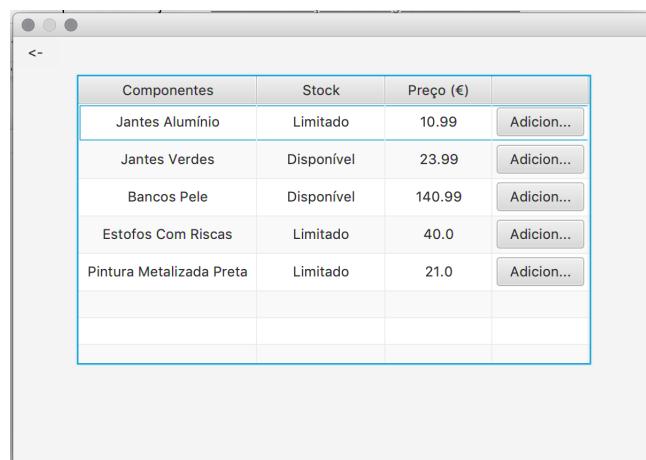


Botão: Abrir

Para visualizar os componentes de um pacote o utilizador tem que clicar no botão abrir, presente na linha do respetivo pacote. Será aberto um novo ecrã com os componentes constituintes do pacote:

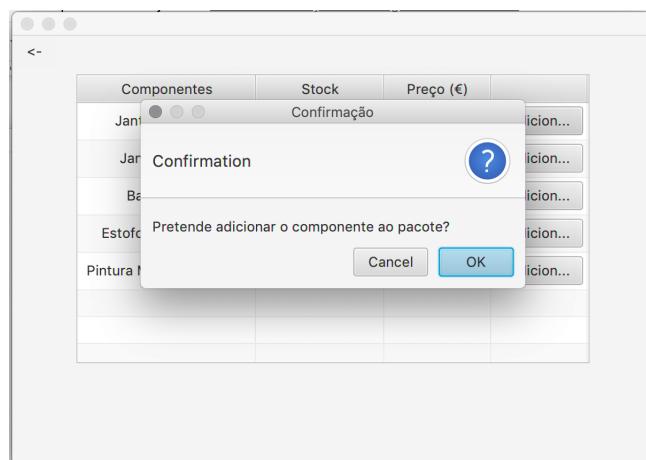


A par dos pacotes, se o administrador clicar no botão apagar, para apagar um componente do pacote, será apresentada uma mensagem de confirmação:

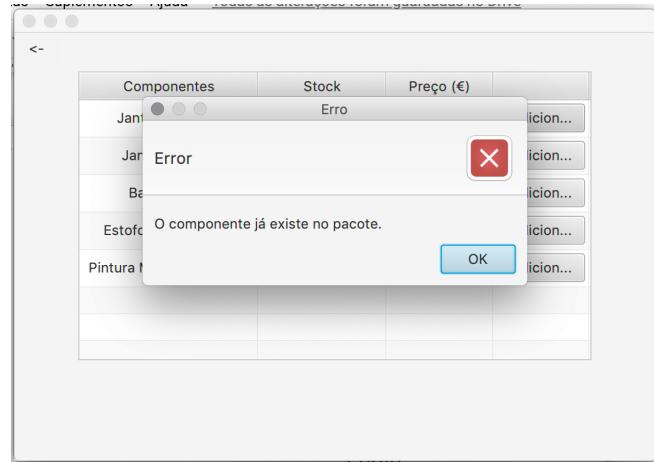


Botão: Componentes

De modo a adicionar mais componentes ao pacote, o administrador terá que clicar no botão Componentes, onde será apresentado um ecrã com todos os componentes existentes no sistema, assim como a opção de adicionar o componente ao pacote.

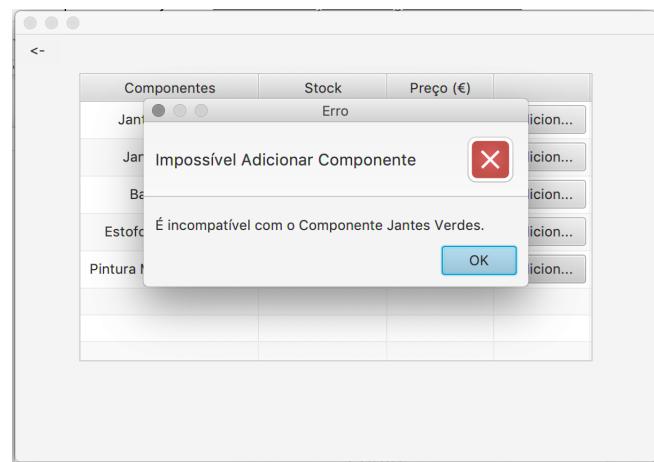


Caso clique no botão para adicionar será apresentada uma mensagem de confirmação:

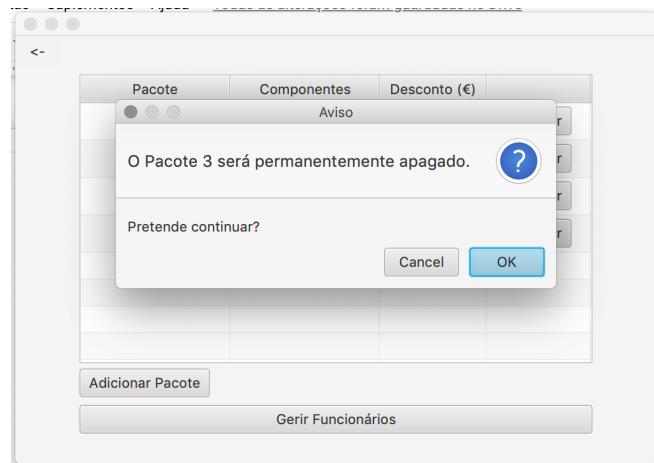


Tendo em consideração o componente escolhido pelo administrador, poderão ser apresentados diferentes avisos de insucesso na operação:

1. Componente já presente no pacote



2. Existe pelo menos um componente incompatível no pacote com o componente que pretende adicionar

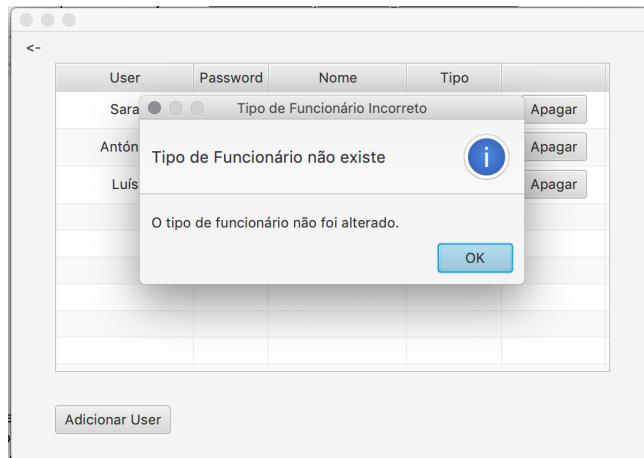


Voltando ao menu do administrador, caso a intenção do administrador seja adicionar/remover/editar funcionários, deverá clicar no botão Gerir Funcionários, onde será apresentada a seguinte janela:

User	Password	Nome	Tipo	
Luís	1	Utilizador 1	Admin	<button>Apagar</button>
António	3	Utilizador 3	Stand	<button>Apagar</button>
Sara	2	Utilizador 2	Fábrica	<button>Apagar</button>

Adicionar User

Neste ecrã o administrador consegue ver as informações de cada um dos utilizadores com permissões para utilizarem a aplicação. Poderá editar tanto o nome do user, como a password ou o tipo. No caso do tipo, que irá influenciar diretamente as funcionalidades do respetivo utilizador na aplicação, caso seja colocado um tipo inválido aparecerá uma mensagem de aviso no ecrã e o tipo de funcionário não será alterado:



Botão: Adicionar User

Para adicionar um novo utilizador o administrador deverá clicar no botão adicionar user. Será adicionado um novo utilizador com as informações default, que podem ser conferidas na figura seguinte. O administrador poderá editar os campos user, password e tipo de qualquer um dos utilizadores.

<-

User	Password	Nome	Tipo	
Luís	1	Utilizador 1	Admin	<button>Apagar</button>
empty	empty	Utilizador 4	Stand	<button>Apagar</button>
Sara	2	Utilizador 2	Fabrica	<button>Apagar</button>
António	3	Utilizador 3	Stand	<button>Apagar</button>

Botão: Apagar

Ao apagar um utilizador também irá aparecer uma mensagem de confirmação.

The screenshot shows a user management interface with a table of users and a confirmation dialog box.

User Table:

User	Password	Nome	Tipo	
empty	● ● ●	Aviso		Apagar
Sara	Confirmation			?
Antón				Apagar
Luis	Pretende continuar?			Apagar

Confirmation Dialog:

Pretende continuar?

Buttons:

- Cancel
- OK

Bottom Navigation:

Adicionar User

Funcionário do Stand

Quando um funcionário do stand inicia sessão na aplicação é confrontado com uma janela com todas as configurações ativas no sistema, tal como se pode ver no exemplo da figura seguinte.

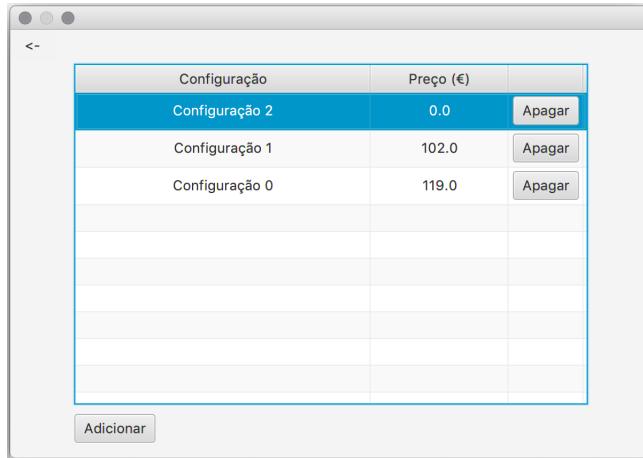
<-

Configuração	Preço (€)	
Configuração 0	119.0	<button>Apagar</button>
Configuração 1	102.0	<button>Apagar</button>

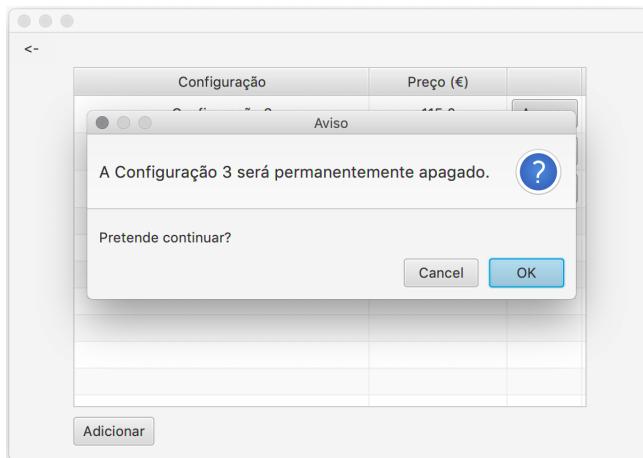
[Adicionar](#)

Botão: Adicionar

Clicando na opção de Adicionar, é adicionada uma nova configuração ao sistema com as informações default, ou seja, com preço nulo e sem componentes ou pacotes.



Botão: Apagar Clicando no botão de apagar, presente na linha da configuração, surge uma mensagem de confirmação no ecrã, onde o utilizador deverá confirmar a pretensão de apagar a configuração.



Configuração

Voltando à Figura do Menu, para se poder consultar todos os detalhes de uma configuração e alterá-la o Funcionário do Stand deverá clicar 2 vezes na configuração que pretende editar.

Consequentemente, irá aparecer um ecrã idêntico ao da figura seguinte, onde na tabela superior se pode consultar os componentes presentes na configuração e na tabela inferior os pacotes presentes na mesma.

Componente	Stock	Preço (€)	
Pintura Azul Escuro	Disponível	30.0	Apagar
Motor Híbrido	Limitado	100.0	Apagar
Pacote	Componentes	Desconto (€)	

No content in table

Componentes Pacotes Config. Ótima
Criar Encomenda

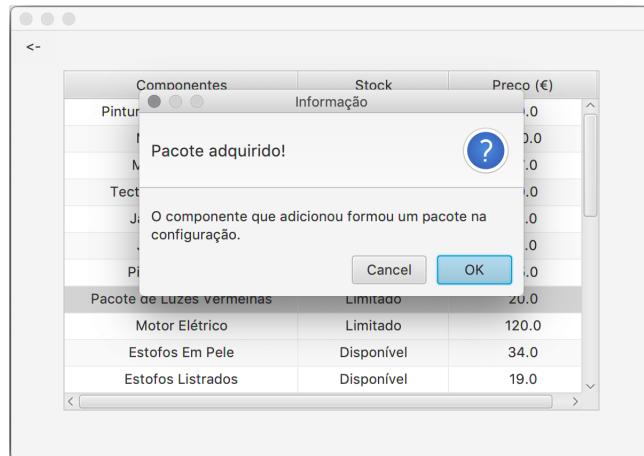
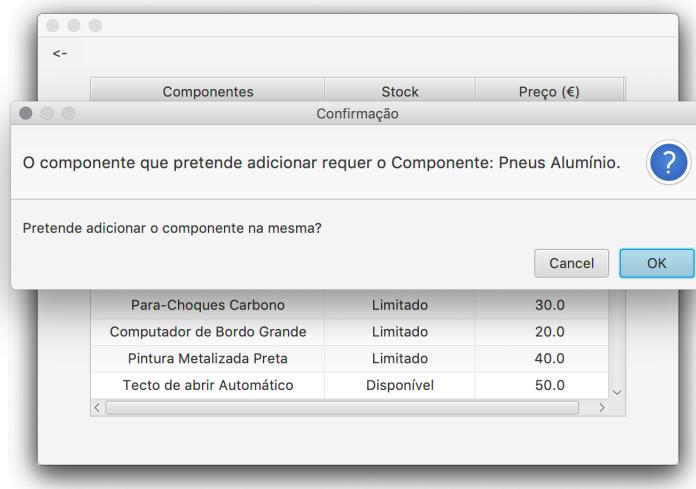
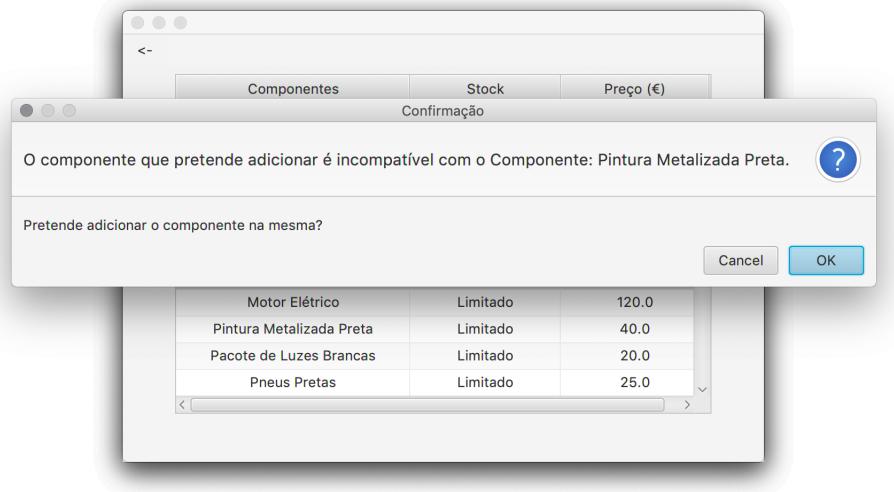
Botão: Componentes

Para adicionar novos componentes à configuração o utilizador deverá clicar no botão Componentes, onde irá aparecer um novo ecrã com todos os componentes que existem no sistema:

Componentes	Stock	Preço (€)
Para-Choques Plástico	Limitado	30.0
Pacote de Luzes Brancas	Limitado	20.0
Computador de Bordo Médio	Limitado	5.0
Estofos Listrados	Disponível	19.0
Pintura Metalizada Preta	Limitado	40.0
Jantes Alumínio	Limitado	10.0
Tecto de abrir Automático	Disponível	50.0
Pneus Pretas	Limitado	25.0
Motor Gasóleo	Limitado	89.0
Motor Híbrido	Limitado	100.0
Vidros Transparentes	Disponível	10.0

Para adicionar um dos componentes basta clicar 2 vezes no componente. Assim, poderá acontecer uma das X coisas:

1. Componente que se pretende adicionar ser incompatible com algum dos componentes já presentes na configuração (Figura x);
2. Componente que se pretende adicionar requer componentes que ainda não estão presentes na configuração (Figura x);
3. Componente ser adicionado com sucesso e o utilizador ser reencaminhado para o ecrã da Figura y, mas agora com o novo componente na respetiva tabela (Figura x);
4. Componente adicionado construir um novo pacote (Figura y) e o utilizador ser reencaminhado para o ecrã da Figura y, mas agora com o novo pacote na respetiva e a tabela dos componentes também se encontrar devidamente atualizada.



The screenshot shows a software window with a table of components at the top. The table has columns for Componente, Stock, Preço (€), and Apagar (Delete). It contains three rows: Pintura Azul Escuro (Stock: Disponível, Price: 30.0), Estofos Listrados (Stock: Disponível, Price: 19.0), and Motor Híbrido (Stock: Limitado, Price: 100.0). Below the table is a message: "No content in table". At the bottom are three buttons: Componentes (highlighted in blue), Pacotes, and Config. Ótima. A "Criar Encomenda" button is also present.

Componente	Stock	Preço (€)	
Pintura Azul Escuro	Disponível	30.0	Apagar
Estofos Listrados	Disponível	19.0	Apagar
Motor Híbrido	Limitado	100.0	Apagar
Pacote	Componentes	Desconto (€)	

No content in table

Componentes Pacotes Config. Ótima
Criar Encomenda

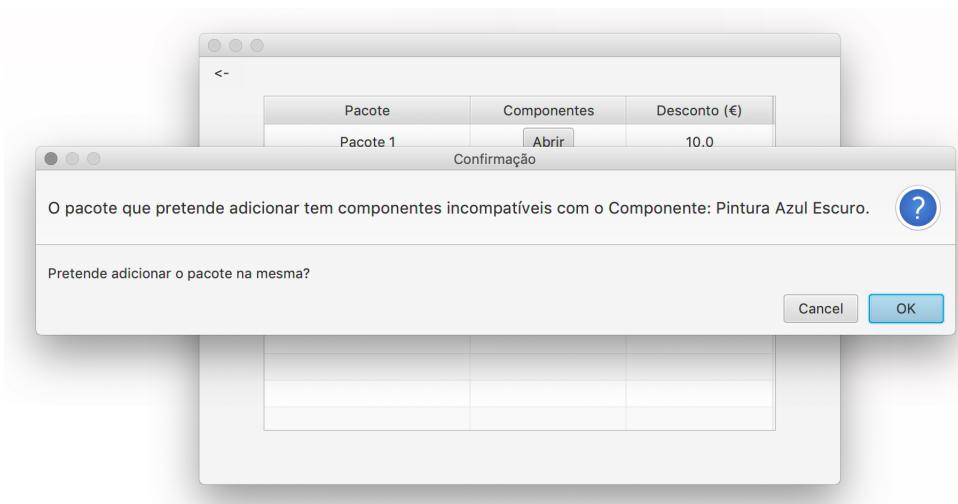
Botão: Pacotes

Novamente a partir da Figura D, clicando no botão Pacotes, o utilizador irá visualizar todos os pacotes que se encontram no sistema (figura seguinte) e, clicando no botão Abrir, poderá ver todos os componentes que constituem o pacote. Para o adicionar basta clicar 2x no mesmo e o utilizador será reencaminhado novamente para o ecrã da configuração devidamente atualizado.

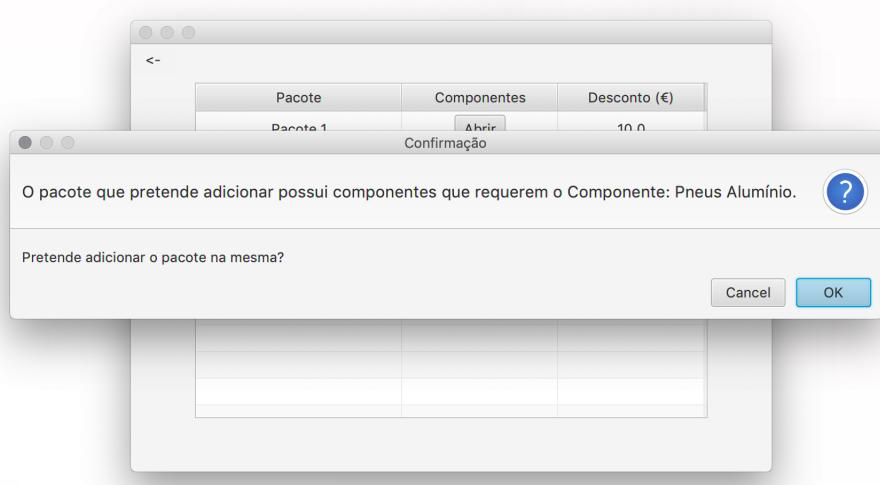
The screenshot shows a software window with a table of packages at the top. The table has columns for Pacote, Componentes, and Desconto (€). It contains three rows: Pacote 2 (Components: Abrir, Desconto: 20.0), Pacote 1 (Components: Abrir, Desconto: 10.0), and Pacote 3 (Components: Abrir, Desconto: 30.0).

Pacote	Componentes	Desconto (€)
Pacote 2	Abrir	20.0
Pacote 1	Abrir	10.0
Pacote 3	Abrir	30.0

O pacote que o utilizador pretende adicionar poderá gerar incoerências de compatibilidade da configuração. Caso o pacote adicionado contenha algum componente que é incompatível com algum dos componentes presentes na configuração aparecerá a mensagem que se pode ver na figura seguinte:



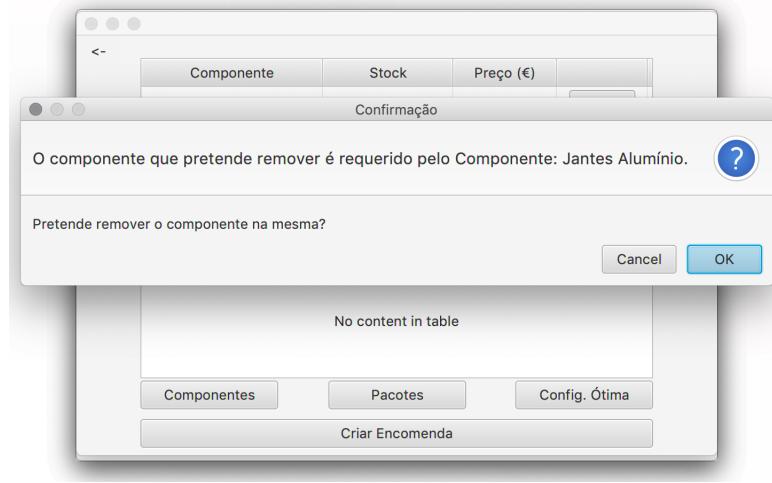
E caso algum dos componentes no pacote requeira um componente que não está presente na configuração aparecerá também uma mensagem:



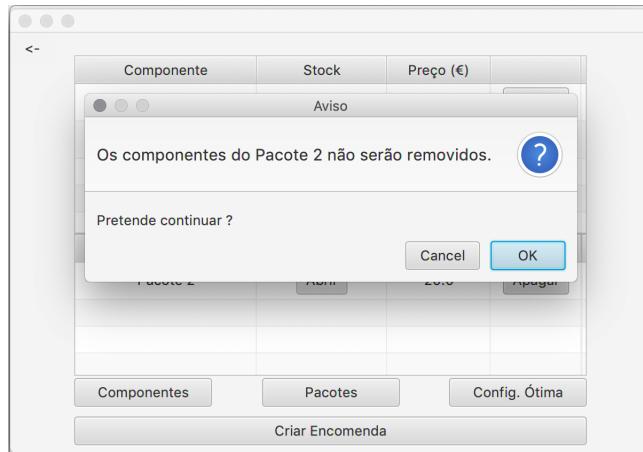
Para que o pacote seja efetivamente adicionado, o utilizador deverá carregar no botão OK. Caso contrário, o pacote não será adicionado.

Apagar Componentes e Pacotes na Configuração

Voltando ao ecrã das informações da configuração (Figura D), para apagar um componente da configuração basta clicar no botão de apagar. No caso do componente removido influenciar algum dos componentes na configuração, isto é, caso seja requerido por algum dos componentes na configuração aparecerá um aviso no ecrã, tal como se pode ver na figura seguinte:



Para apagar um pacote funciona de uma forma um pouco diferente com o intuito de melhorar a UX. Como o utilizador poderá querer apagar somente um componente do pacote, quando o cliente apaga o pacote todos os componentes desse pacote são adicionados à tabela dos componentes (Figura a). Ou seja, os componentes são mantidos mas apenas como componentes individuais, deixando de existir o desconto associado. Assim, o utilizador poderá escolher apagar somente alguns dos componentes e manter os restantes. Caso o utilizador se arrependa da decisão de apagar o pacote e queira voltar a obter o valor do desconto associado ao pack de componentes basta efetuar qualquer operação na aplicação. A aplicação voltará a identificar que se pode constituir um pacote no leque dos componentes da configuração.



Botão: Config. Ótima

Retomando a janela da Figura D, para se poder usufruir da funcionalidade de gerar uma configuração ótima o utilizador deverá clicar no botão Config. Ótima. Consequentemente, abrirá um novo ecrã, tal como o da figura seguinte, onde o utilizador deverá introduzir o valor máximo que o cliente está disposto a gastar.

The screenshot shows a software window with two empty tables. The top table has columns for 'Componentes', 'Stock', and 'Preço (€)'. The bottom table has columns for 'Pacote', 'Componentes', and 'Desconto (€)'. Below the tables is an input field labeled 'Valor Máximo:' containing '200 €'. At the bottom are two buttons: 'Gerar Config. Ótima' and 'Aplicar'.

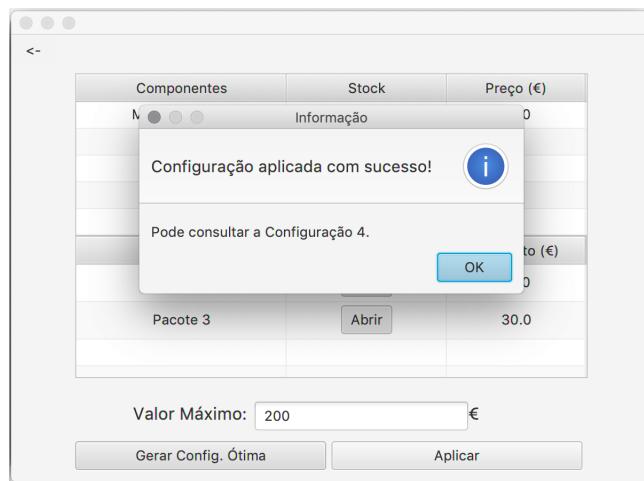
Botão: Gerar Config. Ótima

Posteriormente, o utilizador deverá clicar no botão Gerar Config. Ótima. As duas tabelas serão devidamente atualizadas com os novos componentes/pacotes sugeridos para serem adicionados à configuração, tal como se pode ver na figura seguinte.

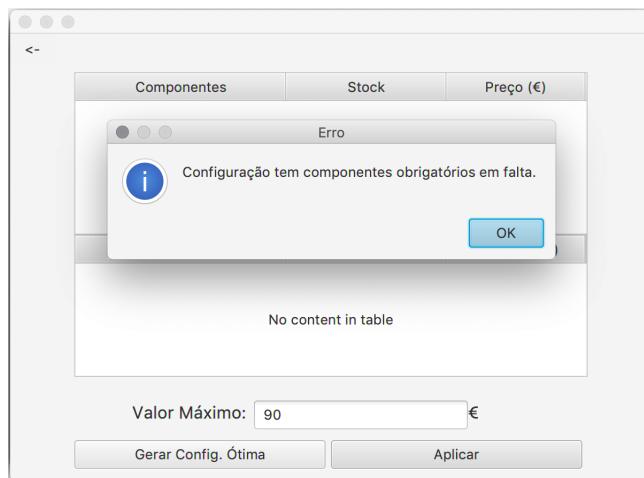
The screenshot shows the same software window after generating the optimal configuration. The two tables now contain suggested components and packages. The input field 'Valor Máximo:' still shows '200 €'. The 'Aplicar' button is visible at the bottom.

Botão: Aplicar

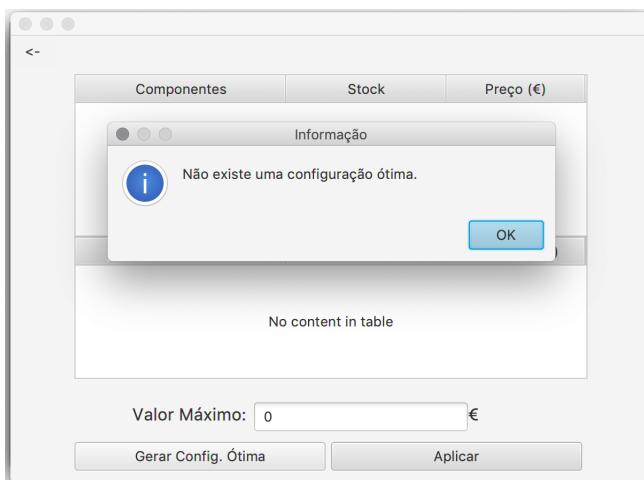
Para aplicar a nova configuração basta clicar no botão Aplicar, aparecendo a mensagem que se pode ver na figura seguinte:



Porém, existem outros cenários possíveis quando se pretende gerar uma configuração ótima. Por exemplo, se a configuração que se pretende otimizar ainda não contiver os componentes básicos (obrigatórios) não será possível obter a configuração ótima, aparecendo a seguinte mensagem:

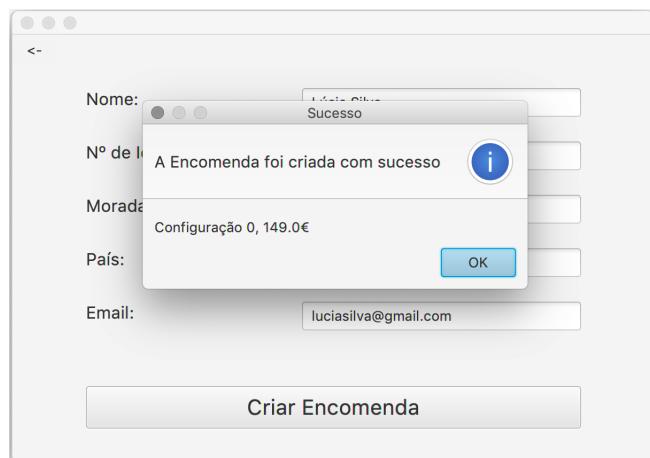
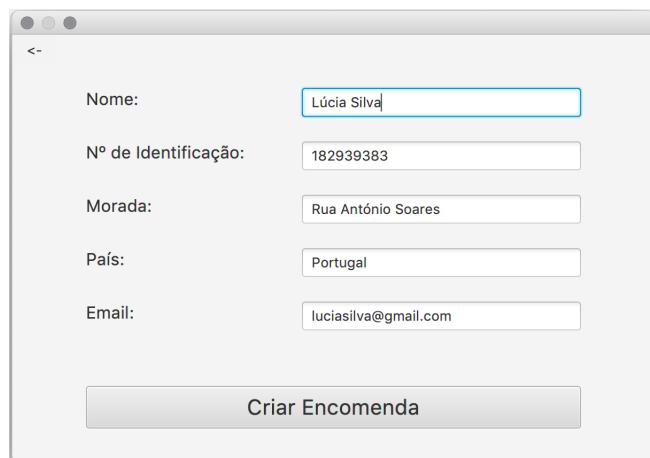


Outra possível opção é o valor introduzido ser demasiado baixo, não sendo possível gerar uma nova configuração. O aviso que irá aparecer no ecrã é o seguinte:



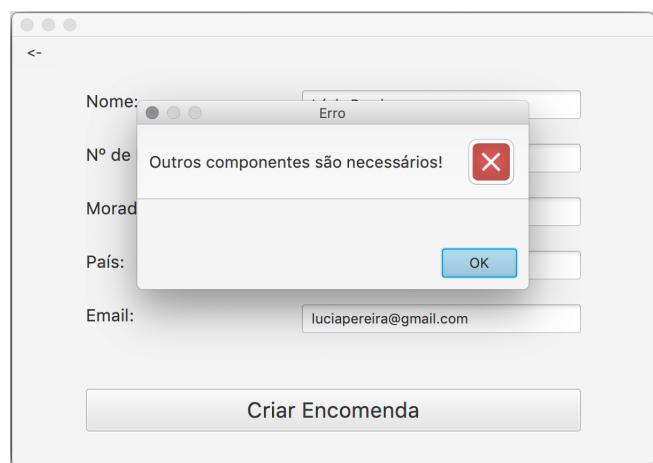
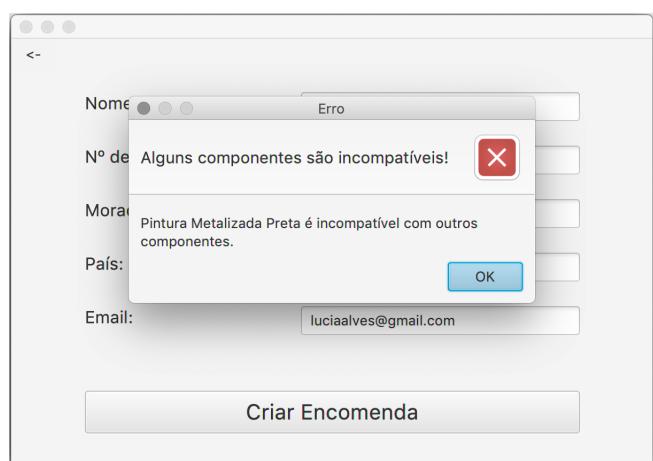
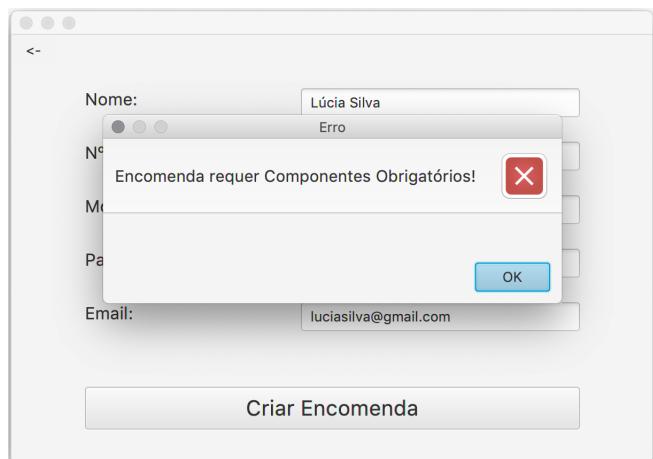
Botão: Criar Encomenda

Retomando a Figura, se o utilizador já estiver satisfeito com a configuração atual poderá criar a respetiva encomenda. Clicando no botão criar encomenda, irá aparecer um ecrã onde devem ser introduzidas todas as informações necessárias para a criação da encomenda. Após todos os dados estarem preenchidos, por exemplo como a Figura, o utilizador deverá clicar no botão Criar Encomenda. Clicando nesse botão a encomenda será efetuada, aparecendo uma mensagem de sucesso tal como se pode ver na Figura.



No entanto, existem alguns possíveis cenários de exceções:

1. A configuração não possui todos os componentes obrigatórios essenciais (Figura X);
2. A configuração possui componentes incompatíveis (Figura X);
3. A configuração possui componentes que requerem outros componentes que não estão presentes na configuração (Figura X).



Funcionário da Fábrica

A última alternativa é o utilizador autenticado se tratar de um funcionário da fábrica. Neste último caso, a janela que irá abrir será a representada na seguinte.

A screenshot of a software application window. At the top, there is a title bar with three circular icons. Below the title bar is a table with three columns: 'Componente', 'Stock', and 'Preço'. The table contains the following data:

Componente	Stock	Preço
Jantes Pretas	4	15.0
Pneus Pretas	9	25.0
Motor Gasolina	6	67.0
Pacote de Luzes Bran...	8	20.0
Pneus Alumínio	5	20.0
Vidros Fumados	50	25.0
Computador de Bordo...	10	5.0
Vidros Transparentes	50	10.0

Below the table, a message reads "Próxima encomenda: 0". At the bottom of the window is a button labeled "Concluir".

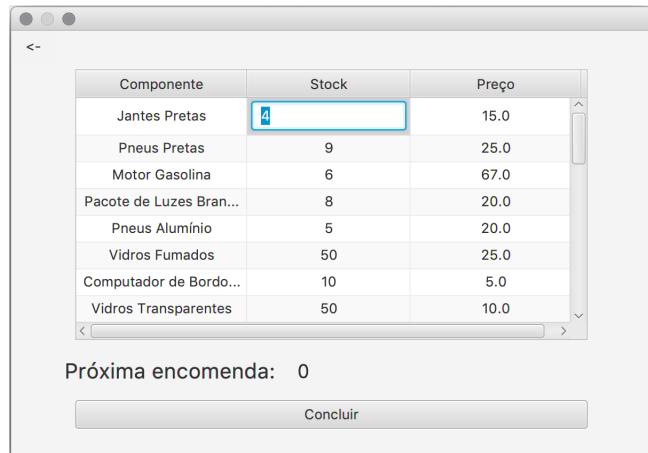
No valor do campo “Próxima encomenda” pode ver-se o id da encomenda que se encontra pronta a ser produzida. Clicando no botão Concluir, o id da encomenda será eliminado da queue, sendo substituído pelo id da próxima encomenda pronta. Caso não exista nenhum encomenda onde todos os componentes existam de momento em stock a janela terá a informação que se pode ver na figura seguinte.

A screenshot of a software application window. At the top, there is a title bar with three circular icons. Below the title bar is a table with three columns: 'Componente', 'Stock', and 'Preço'. The table contains the following data:

Componente	Stock	Preço
Pneus Pretas	9	25.0
Pacote de Luzes Azuis	3	20.0
Vidros Transparentes	50	10.0
Jantes Pretas	2	15.0
Tecto de abrir Manual	23	20.0
Tecto de abrir Automá...	22	50.0
Computador de Bordo...	10	5.0
Vidros Fumados	50	25.0

Below the table, a message reads "Próxima encomenda: Nenhuma encomenda disponível". At the bottom of the window is a button labeled "Concluir".

Para atualizar o stock de algum dos componentes o funcionário da fábrica apenas deverá clicar duas vezes no campo que pretende alterar (tal como se pode ver na figura a). Caso o valor inserido seja inválido (menor que 0) irá aparecer uma mensagem de erro no ecrã (Figura).



Acordamos que para o funcionário visualizar as informações da encomenda terá que utilizar uma outra aplicação, exclusiva para os funcionários da fábrica.

19 Análise Crítica do Resultado Obtido da 1.^a Fase do Projeto

Na parte inicial do projeto, com o intuito de encadear o desenvolvimento deste com base nos componentes, foi concebido através da análise detalhada dos requisitos, um modelo de domínio, de casos de uso e de comportamento da interface do utilizador.

O modelo de domínio é claro e conciso. Esta propriedade, assim como a identificação correta dos diferentes papéis nas relações entre as entidades, vão de encontro aos objetivos que traçamos no começo da concessão deste. Estes objetivos, na nossa apreciação, dão a possibilidade às partes interessadas de compreender e validar as decisões que dizem respeito à implementação, sem que estas tenham as competências técnicas tradicionalmente necessárias para a discussão destes assuntos.

A detalhada especificação dos casos de uso, a parte que consideramos mais desafiante, antecipou problemas não só de insatisfação dos requisitos, como também desmascarou incoerências estruturais nos desenhos iniciais que foram propostos nas várias interações.

O modelo da interface com o utilizador, assegura desde já a conciliação das expectativas dos membros do grupo e, principalmente, das partes interessadas quanto ao resultado gráfico e comportamental do sistema de software que nos comprometemos a desenvolver.

Estes modelos não só facilitaram a convergência do grupo numa ideia unificada para a solução do problema proposto, como também, através da incremental introdução de complexidade nestes, permitiu tornar evidente as relações entre os diferentes componentes do sistema, levando assim à realização deste conforme os requisitos que as partes interessadas estabeleceram. Para lá da simplificação do problema, adicionalmente, estes modelos, catalisaram a criação de um caminho para a implementação com sucesso de um sistema de software que será funcional, satisfará os seus casos de uso e, acima de tudo, será flexível e utilizável.

20 Conclusão Final

A segunda fase do projeto envolveu a exploração dos use cases, identificação das principais classes e packages do projeto e na sua implementação. Nesta fase as principais funcionalidades do projeto já se encontravam definidas por causa disto as decisões que tomamos nesta fase estavam mais ligadas a detalhes que iriam afetar a implementação do sistema do que a detalhes relacionados com o domínio do problema.

O desenvolvimento dos diagramas ajudou a acelerar o processo de desenvolvimento da aplicação que viríamos a desenvolver. No entanto, depois de avançarmos na implementação da aplicação a manutenção destes diagramas passou a ser algo trabalhoso uma vez que o aparecimento de bugs e de comportamentos que não foram previstos eram maiores do que as alterações que poderíamos efetuar.

Uma das decisões mais importantes que tomamos sobre o projeto foi estilo de loading. A metodologia que adotamos no programa foi extremamente lazy, o que nos salva muito espaço em memória. O lado negativo desta aproximação é que, dependendo da distância ao servidor e da quantidade de dados, pode ser muito demorada.

No que toca à configuração ótima a solução atual tem como benefícios de nos dar uma solução ótima ou muito próximo desta e o facto de a solução final poder ser mais facilmente ajustada ao adicionar certas restrições ao modelo. Os problemas que esta solução apresenta são o facto de que requer conhecimentos de programação Inteira e o facto de esta representar um elevado peso computacional o que leva a elevados tempos de resolução quando estamos a trabalhar com um elevado número de componentes e de pacotes.