Documentação de Projeto

para o sistema

Sistema de Gerência de Entregas por Oferta

Versão 5.1

Projeto de sistema elaborado pelo aluno Aylton Bernardino de Almeida Junior e apresentado ao curso de **Engenharia de Software** da **PUC Minas** como parte do Trabalho de Conclusão de Curso (TCC) sob orientação de conteúdo do professor José Laerte Pires Xavier Junior, orientação acadêmica do professor Lesandro Ponciano dos Santos e orientação de TCC II do professor Laerte Xavier

11/05/2022

Tabela de Conteúdo

Histórico de Revisões

Nome	Data	Razões para Mudança	Versão
Entrega 3	07/09/21	Início do documento, definindo seções 2.1, 2.2, 2.3 e 4	1.0
Entrega 4	27/09/21	Inclusão das seções 2.4, 3.1, 3.2 e 3.3	2.0
Entrega 5	12/10/21	Inclusão das seções 3.4, 3.5, 3.6 e 5	3.0
Entrega 6	26/10/21	Inclusão das seções 6 e 7	4.0
Atualização	14/02/22	Adaptação da marca do projeto e do cronograma	4.1
Atualização	05/03/22	Atualização da tela de detalhes da oferta	4.2
Atualização	14/03/22	Atualização do DER	4.3
Atualização	02/03/22	Atualização do DER e Mockup	4.4
Atualização	09/04/22	Adição tela de recarregando rota	4.5
Atualização	11/04/22	Atualizando Mockups	4.6
Atualização	24/04/22	Atualizando diagramas	4.7
Atualização	07/05/22	Atualizando diagramas e fazendo post mortem	5.0
Atualização	11/05/22	Ajustes finais	5.1

1. Introdução

Este documento agrega: 1) a elaboração e revisão de modelos de domínio e 2) modelos de projeto para o sistema de Gestão de Entregas por Ofertas. A referência principal para a descrição geral do problema, domínio e requisitos do sistema é o documento de

especificação que descreve a visão de domínio do sistema. Tal especificação acompanha este documento. Anexo a este documento também se encontra o Glossário.

2. Modelos de Usuário e Requisitos

Esta seção tem como objetivo descrever os usuários e atores do sistema, assim como os requisitos aos quais esse deve atender. Para isso, é apresentada uma breve descrição de cada ator, assim como um modelo desse ator como usuário do sistema. Além disso, são apresentados o diagrama de caso de uso e as histórias de usuários relacionadas, ambos que servem de referência para desenvolvimento do sistema. Por último, são apresentados o diagrama de sequência do sistema e o contrato de comunicações, responsáveis por definir como a aplicação a ser projetada deve se comunicar com aplicações já existentes.

2.1 Descrição de Atores

Fornecedor: Este ator vende produtos por meio da aplicação Web da Zapt. Seu principal objetivo ao utilizar a aplicação proposta é visualizar as ofertas nas quais seus produtos foram vendidos e poder enviá-las aos entregadores que as realizarão.

Entregador: Este ator faz a entrega dos produtos comprados por meio da aplicação Web da Zapt. Seu principal objetivo ao utilizar a aplicação é ter a visualização da rota a ser seguida e quais produtos entregar a cada ponto dessa. Dessa mesma forma ele deseja assinalar pedidos entregues ou com problemas na hora do ocorrido.

Gerente de Operações: Este ator é o gerente do time de operações na Zapt, sendo responsável pela coordenação das ofertas realizadas na empresa. Seu principal objetivo ao utilizar a aplicação proposta é ter uma visão detalhada das entregadas realizadas.

2.2 Modelos de Usuários

Esta subseção tem como objetivo descrever os modelos de usuários desenvolvidos por meio da implementação de personas. Para realização do desenvolvimento das personas que se seguem, foram feitas entrevistas com representantes dos usuários previstos da plataforma.

A Tabela 1 descreve a persona do usuário Sérgio Vieira, um fornecedor de produtos vendidos na Zapt. Nela é possível identificar que esse usuário, por mais que possua

interesse, não se adequa rapidamente à novas tecnologias. Também é perceptível que ele sente falta de uma maneira mais simples de compartilhar as entregas com os entregadores responsáveis. Por fim, após análise é possível ver que ele deseja ter uma melhor organização de suas entregas pendentes, assim como uma melhor forma de compartilhamento delas.

Sérgio Vieira	
Descrição	Sérgio possui 43 anos, nasceu e cresceu em Belo Horizonte e sempre interessou por culinária. Quando criança, aprendeu a receita de pizzas artesanais da família e hoje as produz para venda por meio da plataforma digital da Zapt. Ele gosta de plataformas digitais e se interessa por elas, mesmo não tendo muito tempo para explorá-las. Hoje se responsabiliza pelas entregas da pizza, de forma que possui entregadores responsáveis, com os quais ele entra em contato quando surge uma nova necessidade de entrega, enviando uma planilha contendo a relação entre pedidos a serem entregues e seus endereços.
Dores	• Aprendizado lento em novas plataformas digitais • Necessidade de gerar uma planilha contendo a relação entre pedidos e endereços para os entregadores
Objetivos	• Melhorar a organização das ofertas as quais ele deve fazer as entregas • Compartilhar de maneira mais organizada os pedidos com o entregador

Tabela 1. Persona Sérgio Vieira

A Tabela 2 descreve a persona do usuário Nelson Junior, um entregador de produtos vendidos por meio da Zapt. Sua descrição mostra que ele não tem interesse em tomar mais tempo que o necessário para realizar as entregas que complementam sua renda. Entretanto, devido a necessidade de organizar estas entregas manualmente, ele acaba gastando mais tempo que o desejado. Por fim, ele sente uma dor grande na necessidade de manter uma lista manual dos produtos a serem entregues e em quais endereços, já que isso já o levou a fazer entregas erradas.

Nelson Junior	
Descrição	Nelson possui 53 anos, nasceu e cresceu em Belo Horizonte e sempre trabalha como entregador nas horas vagas servindo de complemento para sua renda mensal. Nunca teve muito interesse em aplicativos a tecnologias mais recentes, sempre usando o mínimo necessário para conseguir fazer seu trabalho. Hoje é um entregador ajudante de Sérgio, recebendo demandas das entregas em formato de planilhas.
Dores	• Dificuldade e falta de interesse no aprendizado de novas tecnologias. • Não gosta de ter que ficar colocando na mão os endereços da planilha em uma aplicação de mapas online para descobrir onde é o endereço. • Acha muito pouco prático ter que manter uma lista manual dos pedidos a serem feitos, quais já foram entregues e quais tiveram problema.
Objetivos	• Não ter que se preocupar muito com a preparação da entrega, desejando apenas pegar os produtos a serem entregues e seguir uma direção dada. • Gastar somente o tempo necessário para fazer as entregas com elas, já que é algo complementar e não sua principal fonte de renda.

Tabela 2. Persona Nelson Junior

A Tabela 3 descreve a persona do usuário Enzo mendes, esse que é gerente do time de operações na Zapt. Tendo em vista sua função, ele acaba tendo pouco tempo para fazer análises detalhadas de entregas realizadas, já que não existe um padrão para como ele recebe dados sobre elas. Desta forma, ele sente uma grande dor ao gerenciar as entregas realizadas, principalmente quando elas ocorrem em Belo Horizonte, por ele morar em São Paulo. Seu principal objetivo é possuir uma visão detalhada e padronizada das entregas a fim de pensar em possíveis melhorias para elas.

Enzo Mendes

Descrição	Enzo possui 32 anos e é gerente do time de operações na Zapt, nasceu e cresceu em São Paulo, tendo se formado em econômica. Seu dia a dia é muito corrido, precisando coordenar diversas pessoas a fim de gerenciar as ofertas que ocorrem diariamente na empresa. Utiliza diariamente diversas aplicações diferentes e sempre estuda sobre novas tecnologias, tendo certa facilidade em aprendê-las.
Dores	• Falta de tempo. • Falta de perspectiva dos problemas relacionados às entregas que ocorrem em Belo Horizonte. • Problema ao gerenciar entregas feitas em outras cidades.
Objetivos	• Ter uma perspectiva mais detalhada das entregas que acontecem diariamente a fim de definir métricas e objetivos para melhorá-las.

Tabela 3. Persona Enzo Mendes

2.3 Modelo de Casos de Uso e Histórias de Usuários

Esta subseção tem como objetivo descrever os casos de uso e histórias de usuário previstos para o projeto. Para isso é apresentado um diagrama de casos de uso onde todos são listados. Dessa mesma forma, também são apresentadas as histórias de usuário relacionadas às funcionalidades previstas para o sistema a ser desenvolvido.

2.3.1 Diagrama de Casos de Uso

A Figura 1 representa o diagrama de casos de uso referente ao sistema proposto. No diagrama é possível ver 3 atores principais, Fornecedor, Entregador e Gerente de operações, esses que são os usuários do sistema. Também é possível visualizar 3 sistemas externos, a Ledger API, Shipping API e Auth API, os dois primeiros são responsáveis por receber e enviar dados necessários para funcionamento da aplicação e o terceiro é responsável por fazer a autenticação do fornecedor na aplicação.

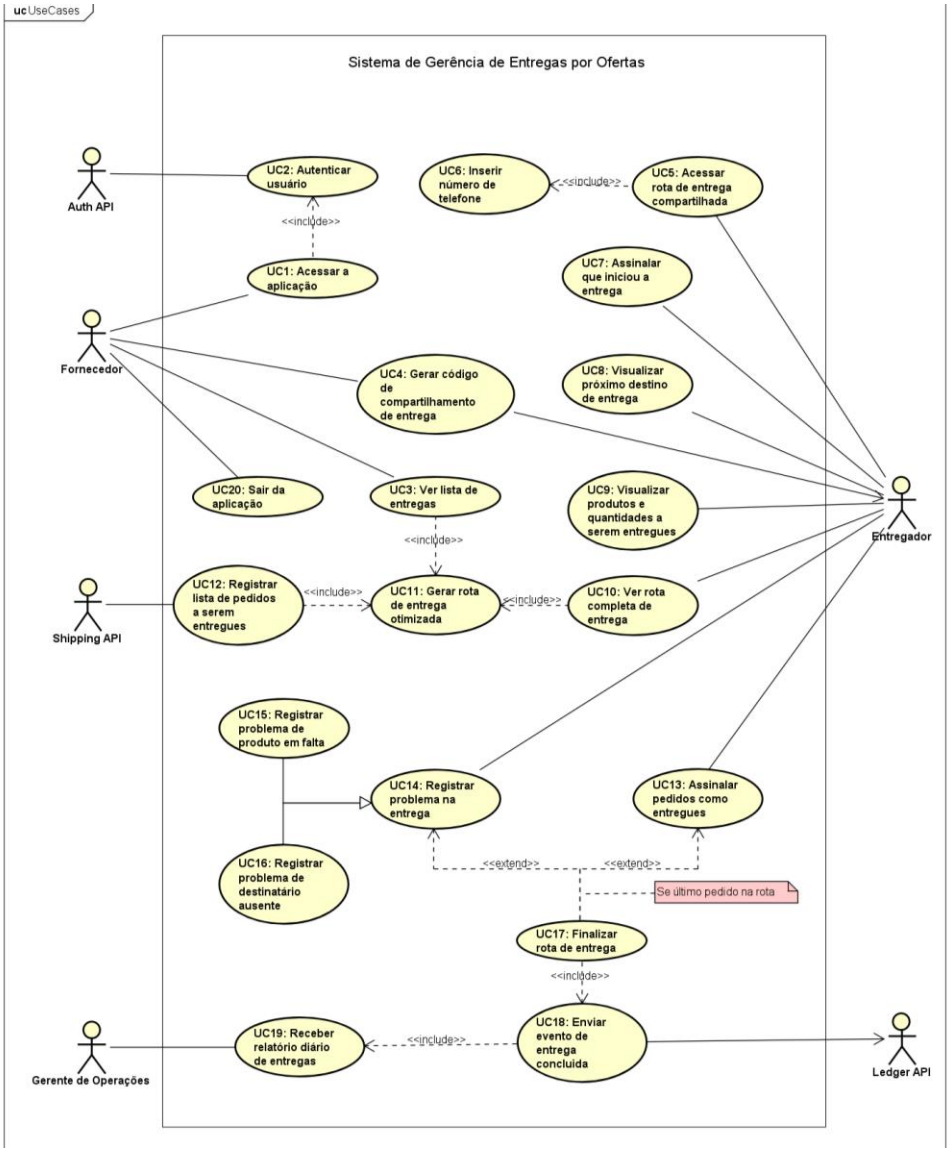


Figura 1. Diagrama de Caso de Uso

2.3.2 Histórias de usuário

US1. Como fornecedor, eu gostaria de ver a lista de entregas pendentes de modo que eu possa alocar entregadores para entregá-las.

US2. Como fornecedor, eu gostaria de gerar um código de compartilhamento de entregas de modo a enviá-lo ao entregador.

US3. Como entregador, eu gostaria de acessar uma rota compartilhada comigo de modo a ver qual a rota de entrega prevista.

US4. Como entregador, eu gostaria de ver a rota completa de entrega de modo a saber quais os pontos de entrega a serem seguidos.

US5. Como entregador, eu gostaria de assinalar que iniciei a entrega de modo a informar o tempo de início da entrega.

US6. Como entregador, eu gostaria de visualizar o próximo destino de entrega de modo a progredir com a entrega de todos os pedidos.

US7. Como entregador, eu gostaria de visualizar quais produtos e suas quantidades devem ser entregues de modo a fazer as entregas corretas em cada endereço.

US8. Como entregador, eu gostaria de assinalar pedidos como entregues de modo a dizer qual o momento exato em que cada entrega foi realizada.

US9. Como entregador, eu gostaria de registrar um problema de entrega quando um produto não está disponível de modo a informar que houve um problema na entrega.

US10. Como entregador, eu gostaria de registrar um problema de entrega quando o destinatário está ausente de modo a informa que houve um problema na entrega.

US11. Como gerente de operações, eu gostaria de receber um relatório diário contendo os dados de entregas realizadas no dia de modo a saber produto, endereços de destino, horas das entregas e entregador responsável.

2.4 Diagrama de Sequência do Sistema e Contrato de Operações

Nesta subseção são apresentados os diagramas de sequência do sistema, assim como seus Contratos de Operações. O objetivo destes diagramas é descrever os fluxos de interação existentes entre os usuários e o sistema a ser desenvolvido.

A Figura 2 representa o diagrama de sequência do sistema relacionado ao fluxo de gerar a rota de entrega otimizada. Neste fluxo, a Shipping API envia uma mensagem assíncrona ao sistema quando uma oferta é finalizada, de forma que a aplicação fica responsável por gerar a rota com base nos pedidos da oferta. Esse diagrama se relaciona aos casos de uso

UC11, UC12, UC3 e UC10, gerando a rota de entrega que será visualizada por ambos fornecedores e entregadores.

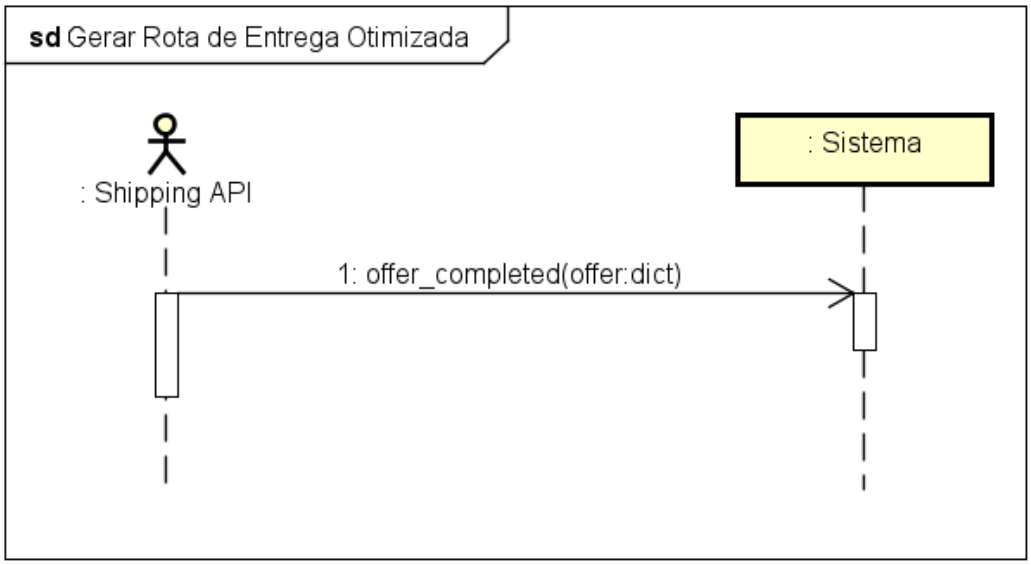


Figura 2. DSS: Gerar rota de entrega otimizada

Contrato	Rota de entrega otimizada
Operação	Offer_copleted(offer: OfferQueue)
Referências cruzadas	Casos de uso: UC3 Ver lista de entregas, UC11 Gerar rota de entrega otimizada, UC12 Registrar lista de pedidos a serem entregues, UC10 Ver rota completa de entrega.
Pré-condições	Oferta deve ter sido finalizada
Pós-condições	A rota de entrega otimizada deve ser armazenada no banco para futuras consultas

A Figura 3 contém o diagrama que modela o fluxo de autenticação do fornecedor. Nele o usuário deve inserir um telefone, de forma que o sistema deve verificar se ele está cadastrado, e depois inserir um código previamente enviado ao seu telefone. Esse fluxo se

relaciona aos casos de uso UC1 e UC2, por representar o processo de entrada do fornecedor na aplicação.

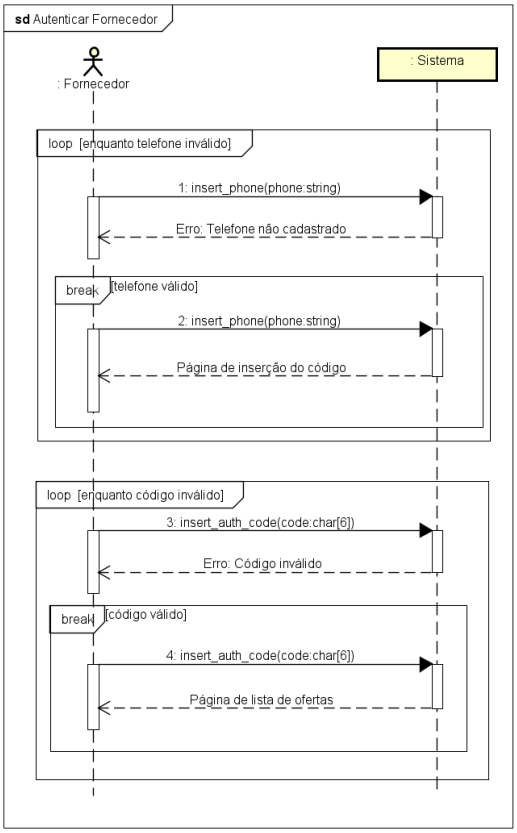


Figura 3. DSS: Autenticar Fornecedor

Contrato	Autenticar Entregador
Operação	Insert_auth_code(code: char[6])
Referências cruzadas	Caso de uso: UC1 Acessar a aplicação e UC2 Autenticar usuário
Pré-condições	Fornecedor deve estar cadastrado na aplicação
Pós-condições	O Fornecedor deve estar autenticado

A Figura 4 representa o fluxo em que um fornecedor deseja compartilhar uma entrega com um entregador. Nele, o usuário acessa a lista de entregas para hoje, vê os detalhes da entrega desejada e seleciona a opção de compartilhá-la com um entregador. Para isso o sistema deve gerar uma mensagem de compartilhamento de forma a enviá-la à um entregador. Esse fluxo se relaciona com o caso de uso UC4, em que o fornecedor deve ser capaz de compartilhar uma entrega com um entregador.

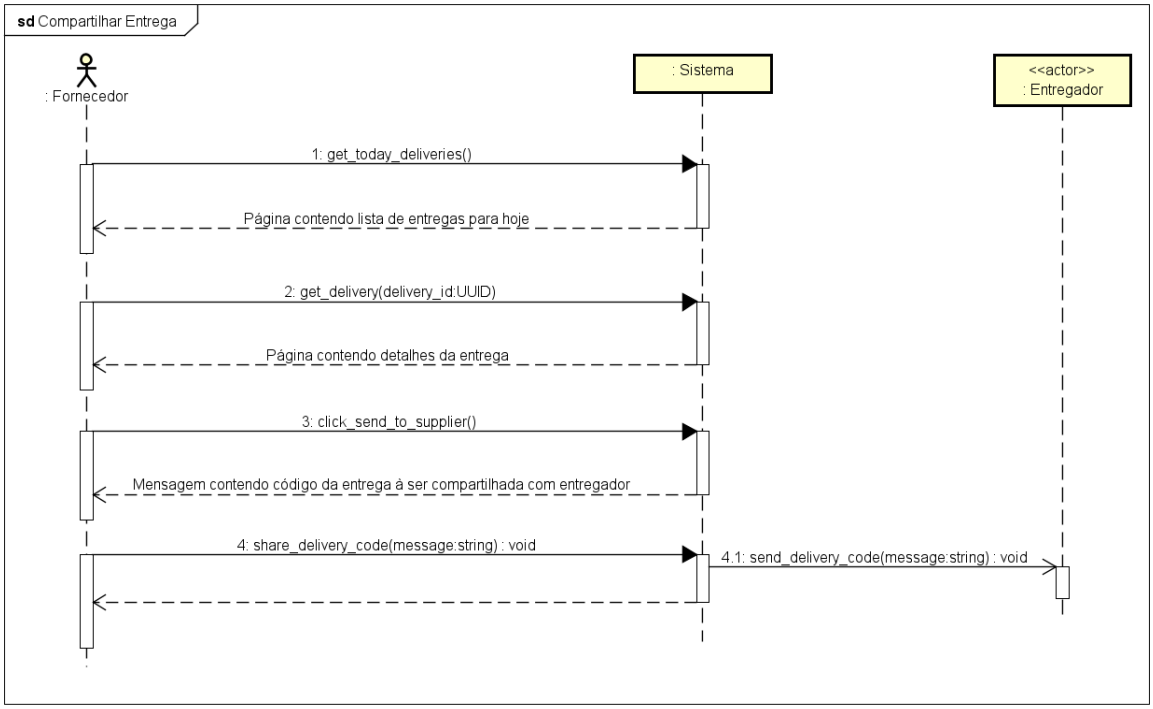
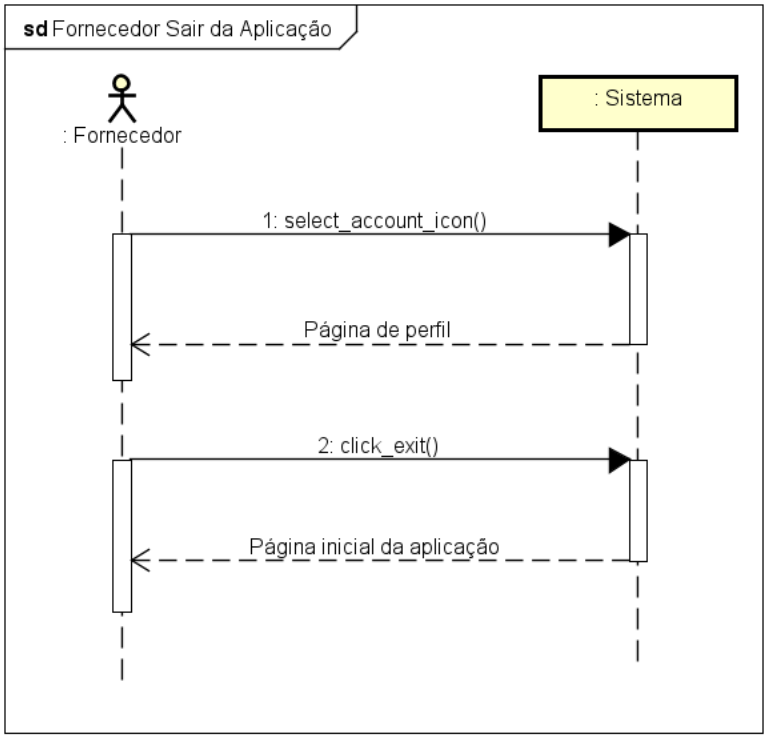


Figura 4. DSS: Compartilhar entrega

Contrato	Compartilhar entrega
Operação	Share_delivery_code(message: string)
Referências cruzadas	Caso de uso: UC4 Gerar código de compartilhamento de entrega
Pré-condições	Fornecedor deve estar autenticado na aplicação

Pós-condições	Um código de acesso à oferta selecionada deve ter sido compartilhado

A Figura 5 representa o fluxo do fornecedor sair da aplicação. Para isso, ele deve acessar seu perfil e selecionar a opção de sair. Esse fluxo se relaciona com o caso de uso UC20, permitindo ao fornecedor sair da conta em que ele está conectado atualmente.



1. *Figura 5. DSS: Fornecedor sair da aplicação*

Contrato	Fornecedor sair da aplicação
Operação	Clieck_exit()
Referências cruzadas	Caso de uso: UC20 Sair da aplicação
Pré-condições	Fornecedor deve estar autenticado

Pós-condições	Fornecedor não deve estar autenticado

A Figura 6 representa o fluxo de autenticação de um entregador. Para isso, ele deve inserir o código de acesso recebido do fornecedor por meio do diagrama representado pela Figura 4. Após inserção de um código válido, ele deve inserir seu telefone para que ele possa ser identificado posteriormente. Esse fluxo se relaciona com os casos de uso UC5 e UC6, por permitirem ao entregador acessar a aplicação ao mesmo tempo que solicitam a ele inserir seu número de telefone.

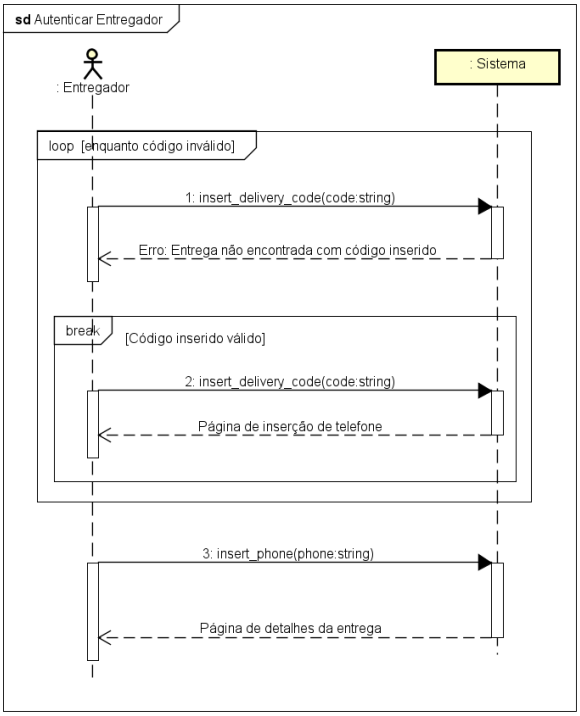


Figura 6. DSS: Autenticar entregador

Contrato	Autenticar entregador
Operação	Insert_phone(phone: string)
Referências cruzadas	Caso de uso: UC5 Acessar rota de entrega compartilhada e UC6 Inserir número de telefone

Pré-condições	Entregador deve ter recebido um código de acesso à uma entrega
Pós-condições	Entregador consegue ver rota de entrega e seus detalhes

A Figura 7 representa o fluxo de entrega de pedidos pelo entregador. Para isso, o usuário deve selecionar a opção de iniciar a entrega e confirmar sua decisão, a partir deste momento ele entra em uma estrutura de repetição para as seguintes etapas até que acabem todos os pedidos:

1. Ver detalhes do pedido a ser entregue
2. Marcar pedido como entregue ou registrar um problema relacionado a ele

Após finalizar a entrega, ele é apresentado com a página de entrega concluída. Este fluxo se relaciona com os casos de uso UC7, UC8, UC9, UC10, UC13, UC14, UC15, UC16 e UC17, representando o fluxo principal de interação com o sistema.

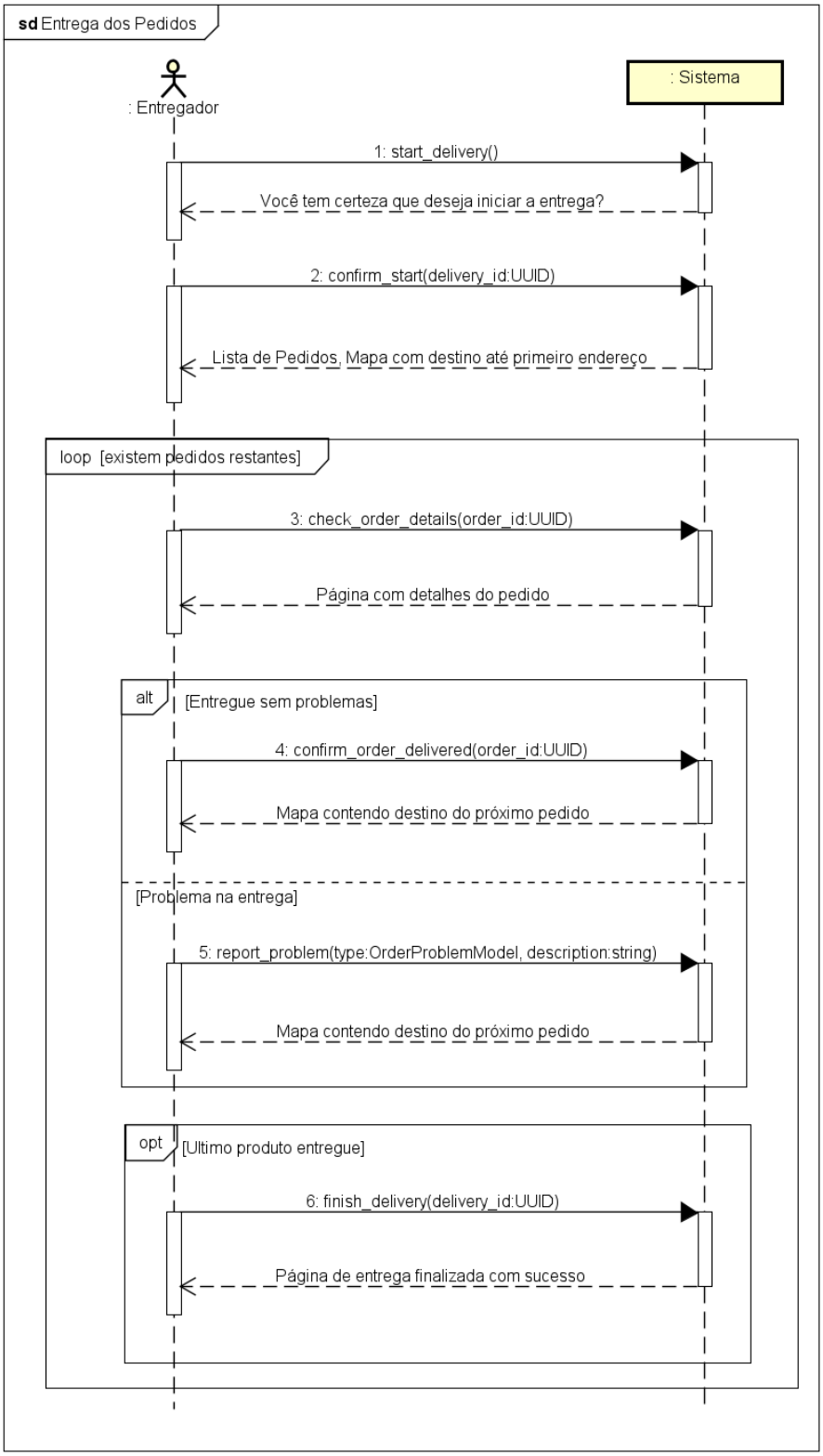


Figura 7. DSS: Entrega de pedidos

Contrato	Entrega de pedidos
Operação	Finish_delivery(delivery_id: UUID)
Referências cruzadas	Casos de uso: • UC7 Assinalar que iniciou a entrega • UC8 Visualizar próximo destino de entrega • UC9 Visualizar produtos e quantidades a serem entregues • UC10 Ver rota completa de entrega • UC13 Assinalar pedidos como entregues • UC14 Registrar problemas na entrega • UC15 Registrar problema de produto em falta • UC16 Registrar problema de destinatário ausente • UC17 Finalizar rota de entrega
Pré-condições	Entregador deve estar autenticado na aplicação
Pós-condições	Entrega deve estar assinalada como finalizada e seus pedidos como entregues.

A Figura 8 representa o fluxo de envio de um relatório de entregas diário ao gerente de operações. Para isso o sistema deve processar a cada 24 horas quais entregas foram realizadas, mas ainda não foram incluídas em um relatório e enviá-las ao gerente de operações. Esse fluxo se relaciona com o caso de uso UC19, que consiste no recebimento de um relatório diário de entregas por parte do ator citado.

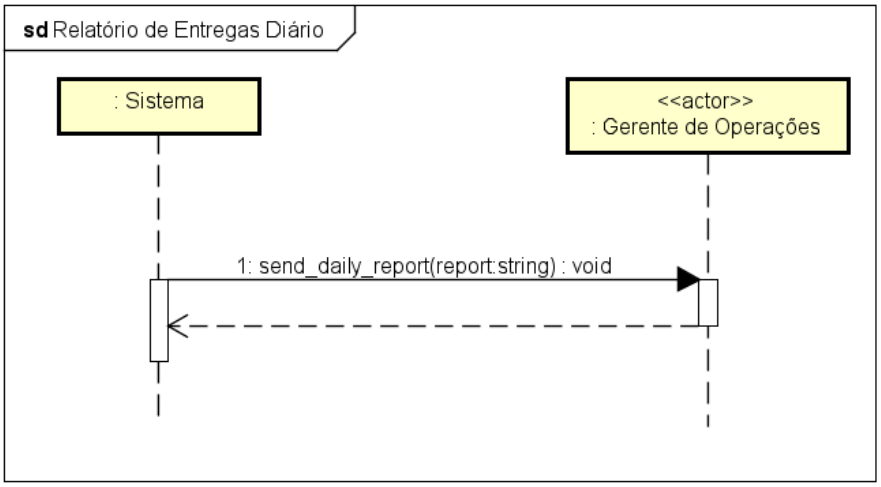


Figura 8. DSS: Relatório de Entregas Diário

Contrato	Relatório de Entregas Diário
Operação	Send_daily_report(report: string)
Referências cruzadas	Caso de uso: UC19 receber relatório diários de entregas
Pré-condições	Novas entregas devem ter sido realizadas nas últimas 24 horas
Pós-condições	O gerente de operações deve receber o relatório de operações

3. Modelos de Projeto

Nesta seção, é apresentada a modelagem dos objetos e fluxos que compõe o sistema proposto. Para isso, são desenvolvidos os seguintes diagramas:

- Diagrama de classes: Tem como objetivo descrever quais classes compõem o sistema e como elas se relacionam.
- Diagrama de sequência: Tem como objetivo descrever quais os fluxos existentes na aplicação, dando foco mensagens trocadas entre os atores e pacotes da aplicação.
- Diagrama de comunicação: Da mesma forma que o diagrama de sequência, esses diagramas têm como objetivo mapear os fluxos existentes na aplicação, porém, tendo o foco nos atores envolvidos.

- Diagrama de arquitetura: Tem como objetivo descrever a arquitetura da aplicação por meio de seus pacotes e como eles se relacionam.
- Diagrama de estados: Tem como objetivo representar a mudança de estados dos objetos com estados maleáveis.
- Diagrama de componentes e implantação: Tem como objetivo descrever como os diversos componentes da aplicação se comunicam e como eles devem ser implantados a nível de camada física.

3.1 Diagrama de Classes

Nesta seção, são apresentados os diagramas de classes que modelam o sistema a ser implementado. Para isso, todos os pacotes que compõem o sistema forma desenhados, representando suas classes e as relações existentes entre elas. O objetivo desta Seção é descrever todos os modelos de dados que são implementados para promover o funcionamento correto da aplicação.

A Figura 9 representa o pacote de *controllers* necessários para funcionamento correto da aplicação. Essas classes são responsáveis por conter as rotas necessárias para que a comunicação do cliente da aplicação seja realizada com a API (do inglês Application Programming Interface). Nele é possível ver três classes, a que contêm as rotas relacionadas às entregas, aos pedidos e ao fornecedor.

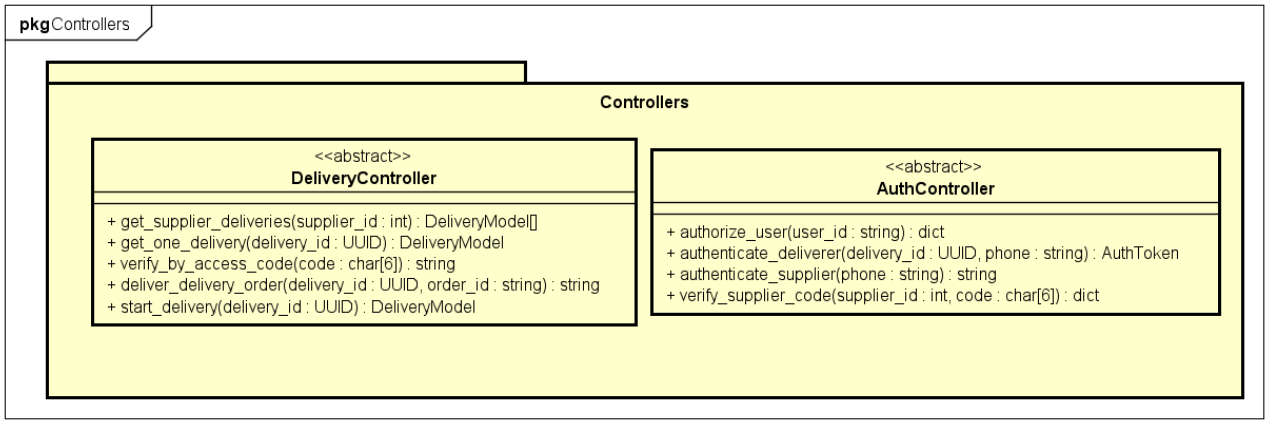


Figura 9. Diagrama de Classes: Controllers

A Figura 10 representa o pacote de *services* necessários para funcionamento da aplicação. Essas classes são responsáveis por fazer o intermédio entre os *controllers* e as demais

classes da aplicação. Nela é possível ver quatro classes relacionadas às entregas, rotas de entregas, pedidos e fornecedores.

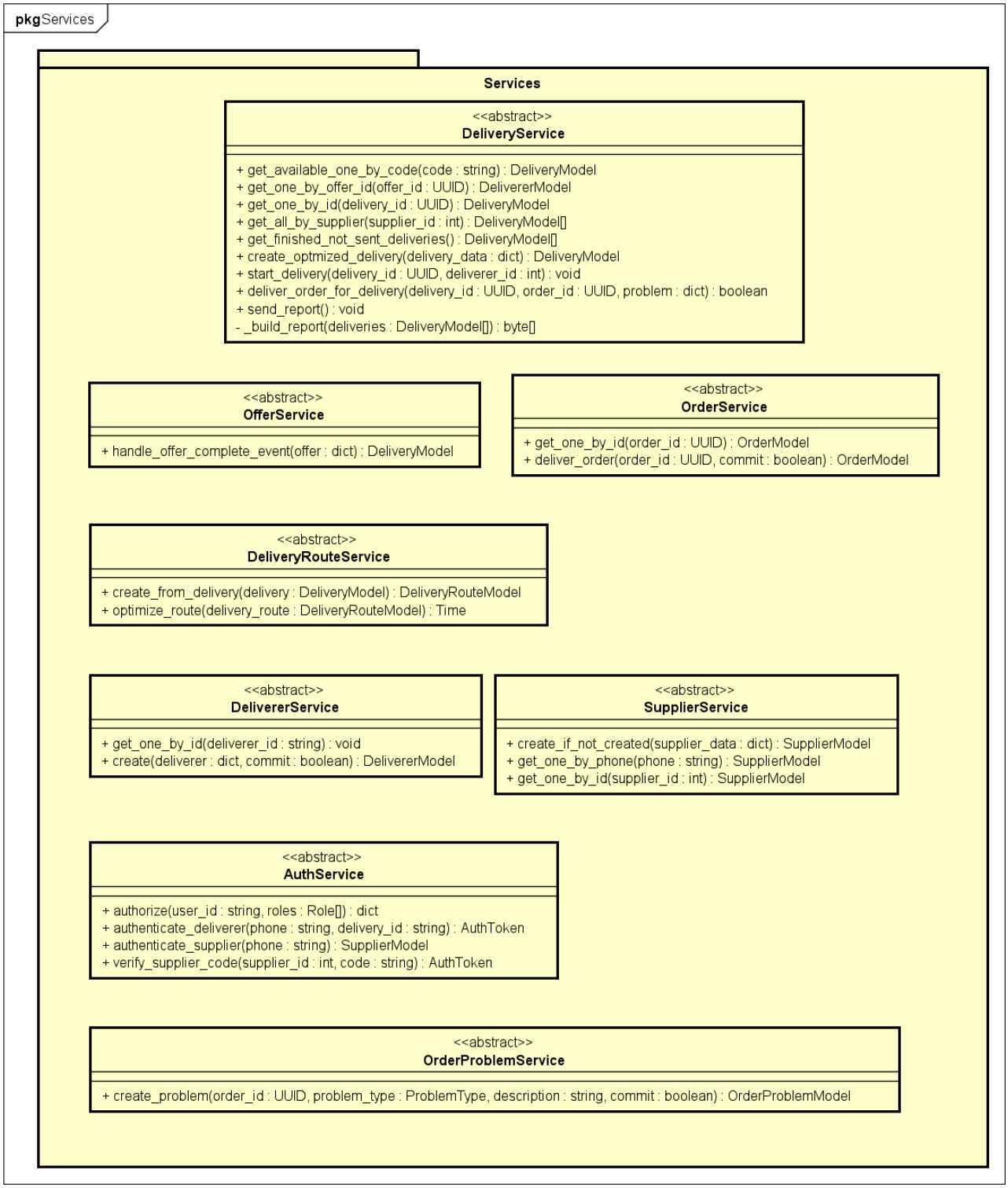


Figura 10. Diagrama de Classes: Services

A Figura 11 representa o diagrama de classes do pacote de modelos de dados a serem implementados no sistema. Nela são mapeadas as classes que vão representar cada modelo necessário para compor a aplicação. Com exceção dos dois *enums* representados, todas as classes herdam da BaseModel, essa que define o método de salvar e atualizar os dados de um modelo no banco de dados. É relevante notar que os dois principais atores da aplicação, fornecedor e entregador, são representados respectivamente pelas classes SupplierModel e DelivererModel.

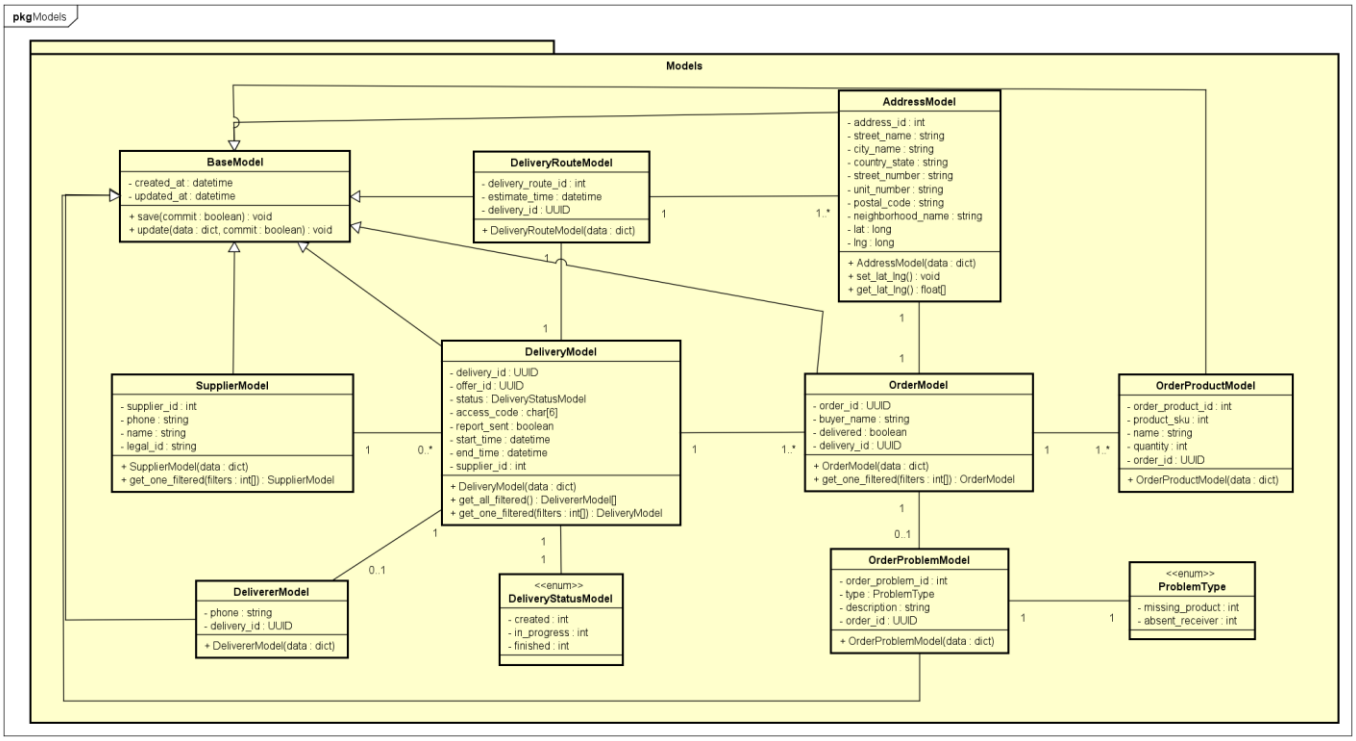


Figura 11. Diagrama de Classes: Models

A Figura 12 representa o pacote de APIs da aplicação. Essas classes são responsáveis por permitir a conexão do sistema a ser desenvolvido com sistemas externos a ele. Tendo este contexto em mente, o pacote possui apenas a classe relacionada à Auth API, que permite ao sistema autenticar usuários por meio deste sistema externo.

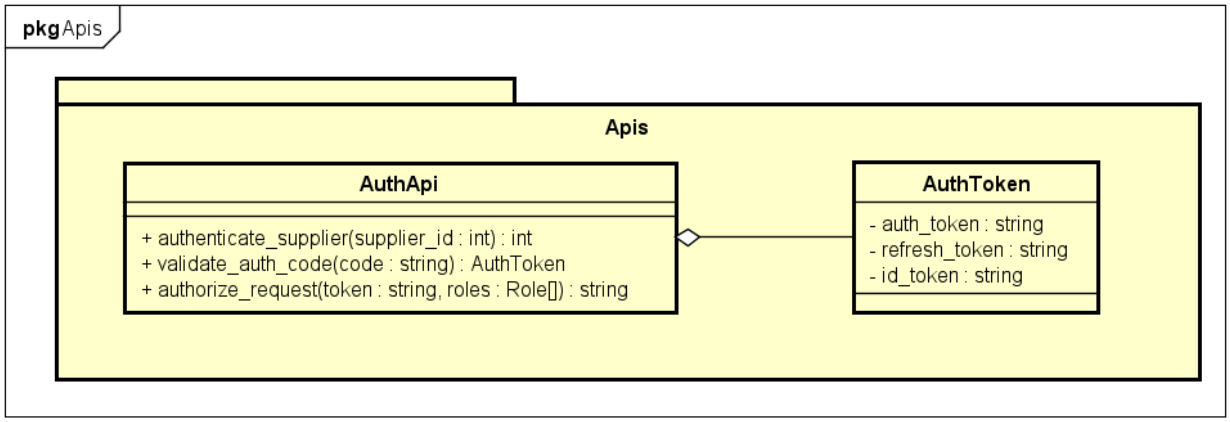


Figura 12. Diagrama de Classes: Apis

A Figura 13 representa o pacote de *events* da aplicação. Esse conjunto de classes tem a responsabilidade de ouvir eventos externos e enviar eventos à outras APIs por meio de um sistema de mensageria. Para isso foram definidas uma fila de mensagens para ouvir os eventos relacionados a completude uma oferta e uma classe que define a mensagem relacionada a finalização da entrega de uma respectiva oferta.

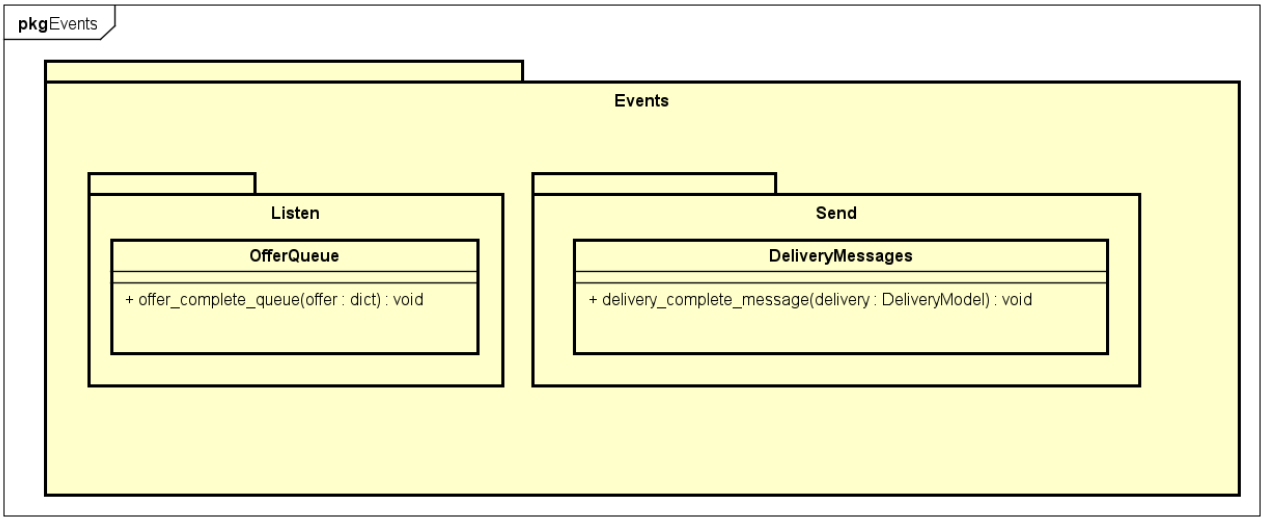


Figura 13. Diagrama de Classes: Events

A Figura 14 representa o pacote de *Jobs* da aplicação. Este pacote contém classes relacionadas a serviços que devem ser executados junto da aplicação. No caso, esse

serviço é o de entregas, que deve ser executado todo dia com a finalidade de enviar o relatório de entrega diário.

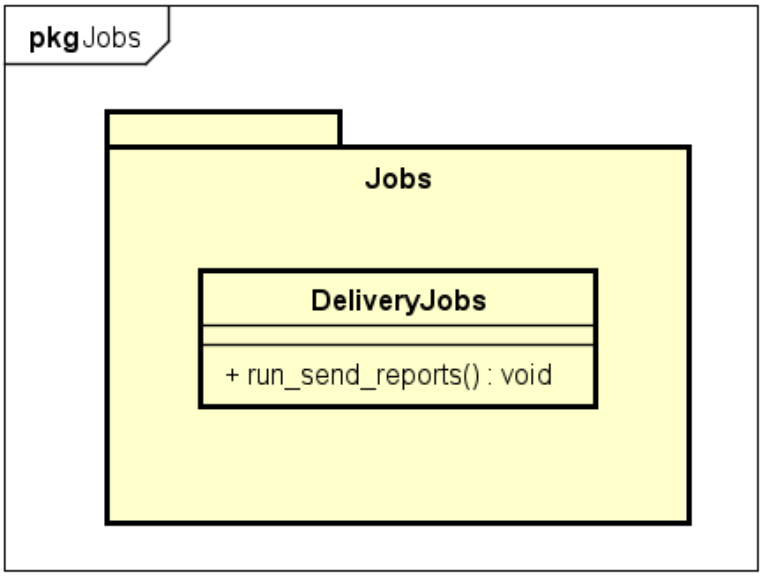


Figura 14. Diagrama de Classes: Jobs

A Figura 15 representa o pacote de *Utils* da aplicação. Este pacote contém classes utilitárias que podem ser usadas pelas demais entidades da aplicação. Nela se encontra a classe XLSXBuilder, essa que tem como objetivo prover a funcionalidade necessária para construção de um arquivo Excel que será enviado ao gerente de projetos.

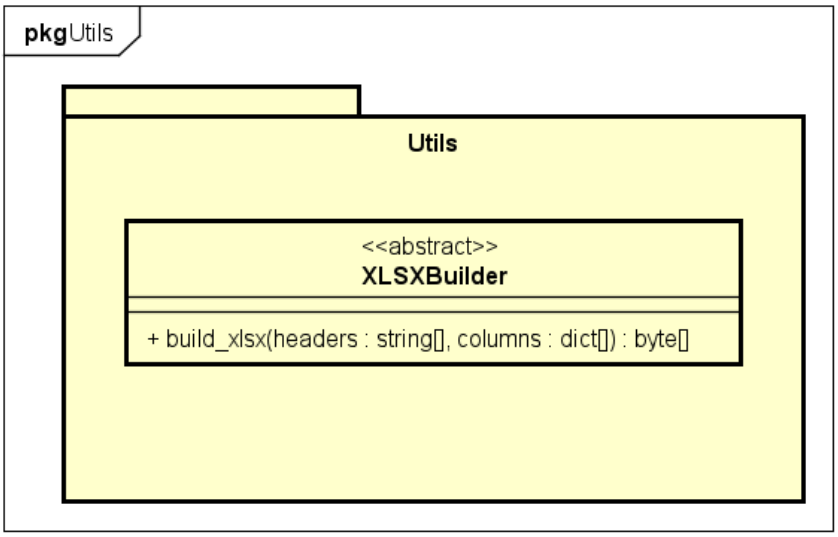


Figura 15. Diagrama de Classes: Utils

A Figura 16 representa o pacote de *Libs* da aplicação. Este pacote contém classes responsáveis por conectar o sistema a bibliotecas externas. Dessa forma, nele está contida a classe MapsAPI, responsável por permitir a comunicação da aplicação com a API do Google Maps, esta que será usada para otimizar a rota de entrega dos pedidos.

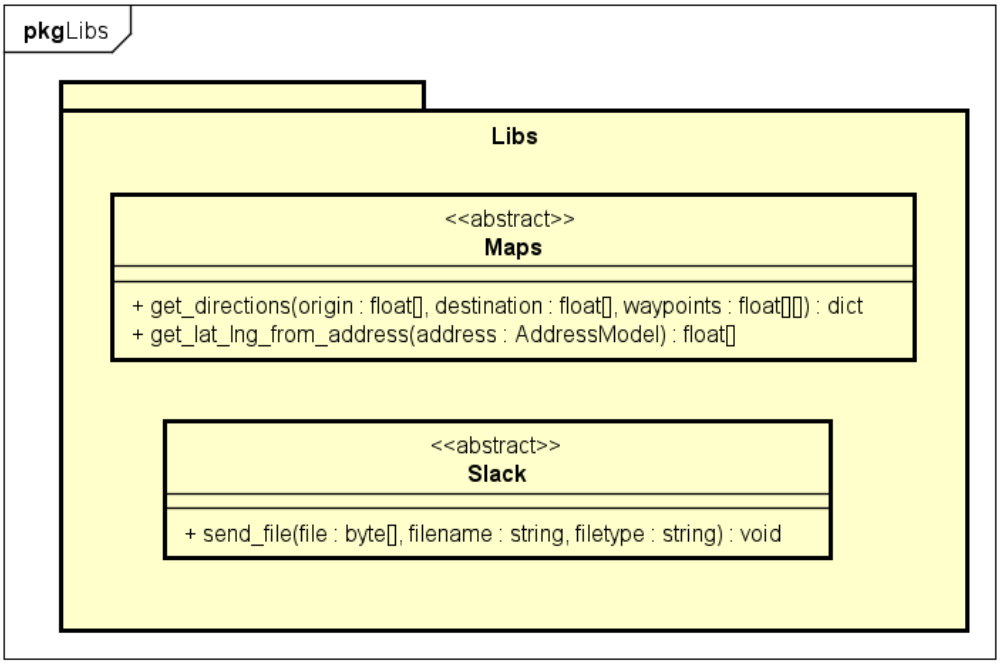


Figura 16. Diagrama de Classes: Libs

3.2 Diagramas de Sequência

Nesta seção, são apresentados os diagramas de sequência que modelam o sistema a ser implementado. Esses diagramas têm como objetivo mapear os principais fluxo seguidos pelos usuários ao utilizar a aplicação. Para isso, as mensagens trocadas entre os diversos pacotes são representadas, assim como o tempo e ordem necessária para que a aplicação funcione da maneira correta.

A Figura 17 representa o fluxo de gerar rotas de entregas otimizadas para o sistema. Esse fluxo consiste no recebimento de uma mensagem assíncrona enviada pela Shipping API. Esta mensagem deve trazer consigo os dados relacionadas à oferta recém finalizada e seus pedidos, para que o sistema consiga produzir uma rota otimizada para as entregas e armazenar essa rota no banco de dados para consultas futuras. Este diagrama serve como

detalhamento do diagrama representado pela Figura 2 e se relaciona aos casos de uso UC11, UC12, UC3 e UC10, gerando a rota de entrega que será visualizada por ambos fornecedores e entregadores.

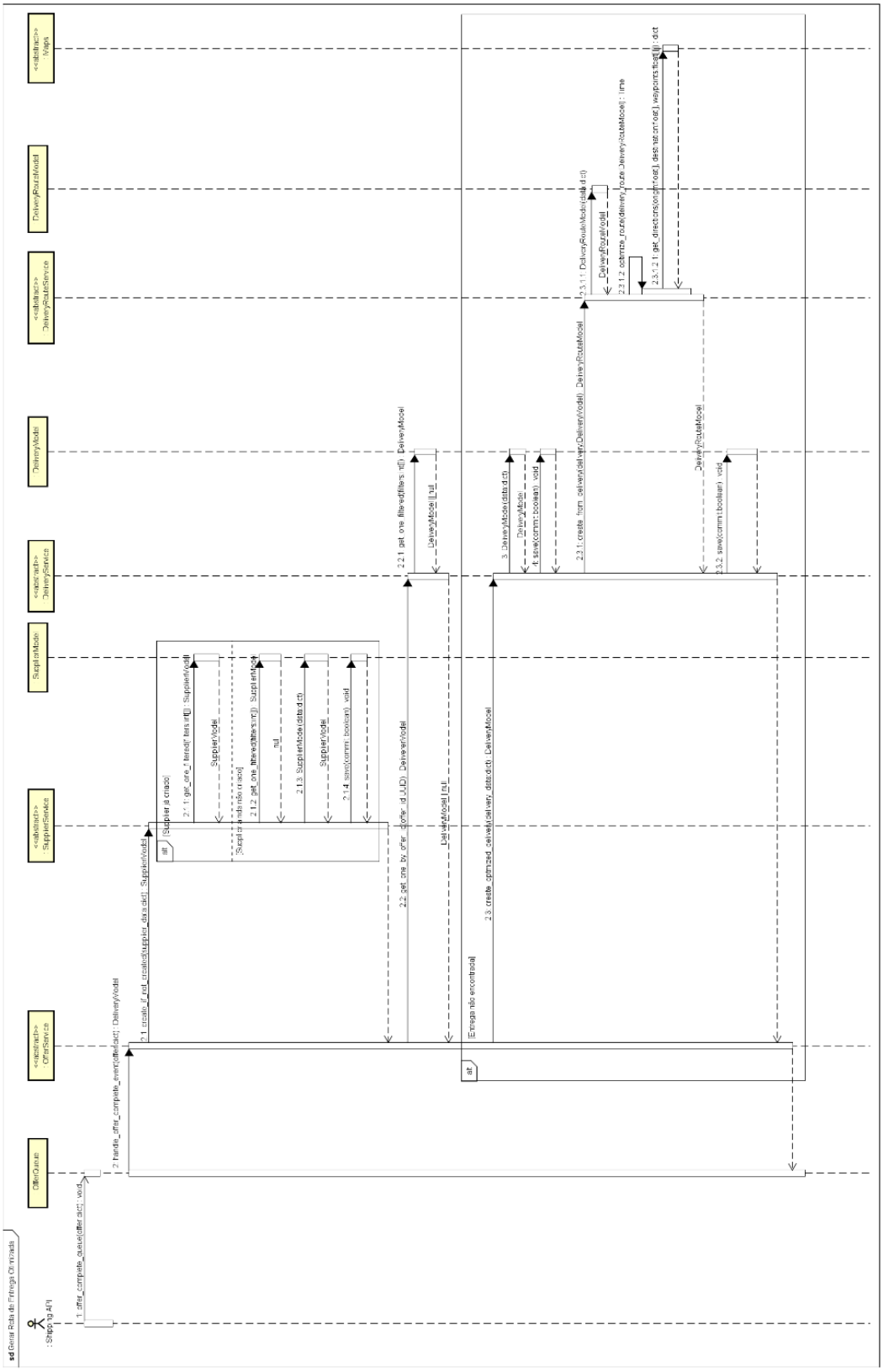


Figura 17. Diagrama de Sequência: Gerar rota de entrega otimizada

A Figura 18 representa o fluxo de autenticação do fornecedor. É possível ver que o fluxo é quebrado em duas partes fundamentais, a validação do telefone inserido, de forma a verificar se ele foi cadastrado previamente, e a validação do código enviado ao telefone inserido. Ambas as verificações são feitas por meio da Auth API, API externa ao sistema sendo projetado. Este diagrama serve como detalhamento do diagrama representado pela Figura 3 e se relaciona aos casos de uso UC1 e UC2, por representar o processo de entrada do fornecedor na aplicação.

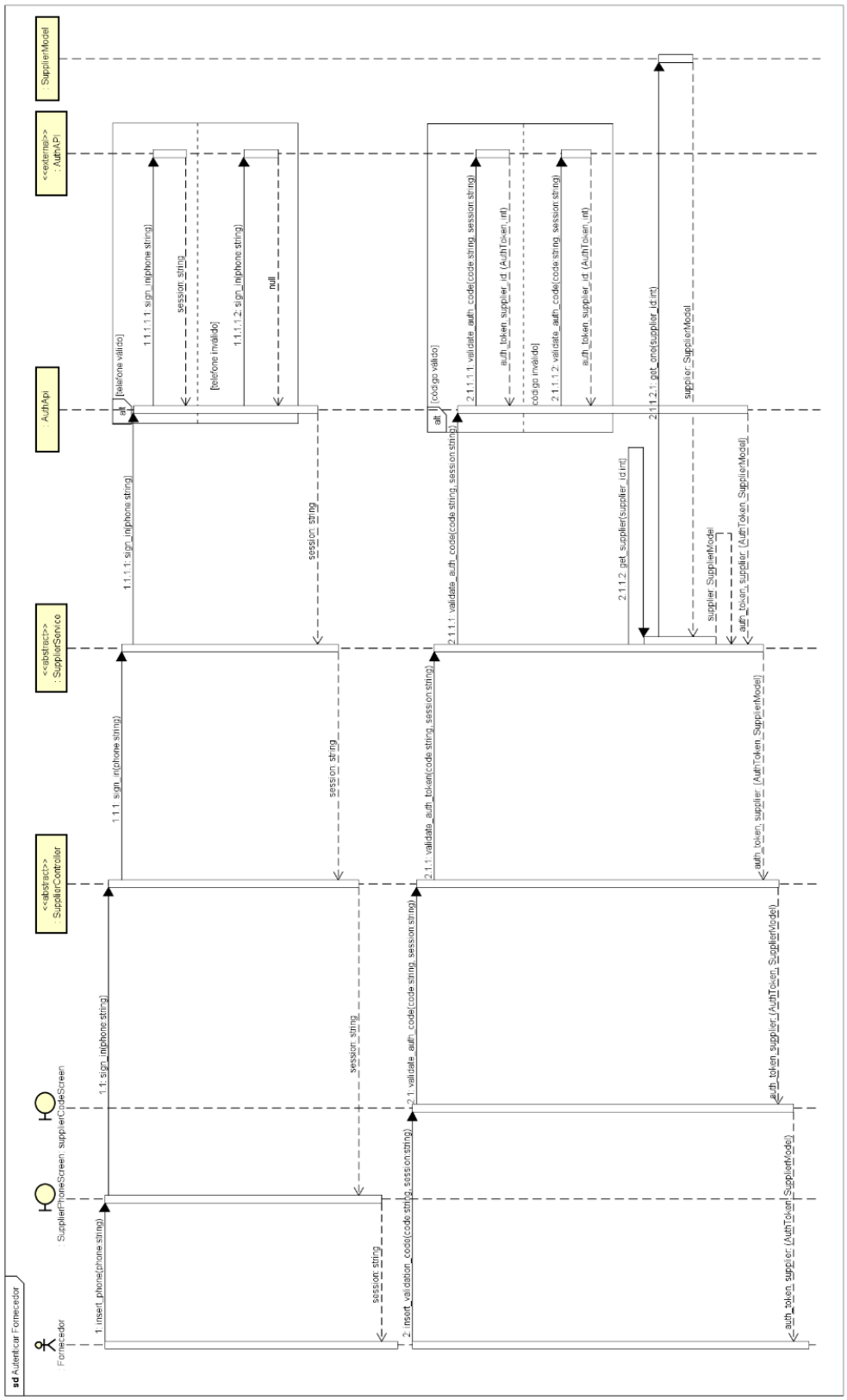


Figura 18. Diagrama de Sequência: Autenticar fornecedor

A Figura 19 representa o fluxo de compartilhamento de uma entrega com um entregador. Para isso, o fornecedor deve selecionar uma oferta, dentre as apresentadas em uma lista de ofertas, e selecionar a opção de compartilhamento da entrega. Dessa forma, o sistema gera uma mensagem de compartilhamento e usa da API ShareIntent para compartilhar a mensagem por meio do meio de comunicação selecionado pelo usuário. Este diagrama serve como detalhamento do diagrama representado pela Figura 4 e se relaciona com o caso de uso UC4, em que o fornecedor deve ser capaz de compartilhar uma entrega com um entregador.

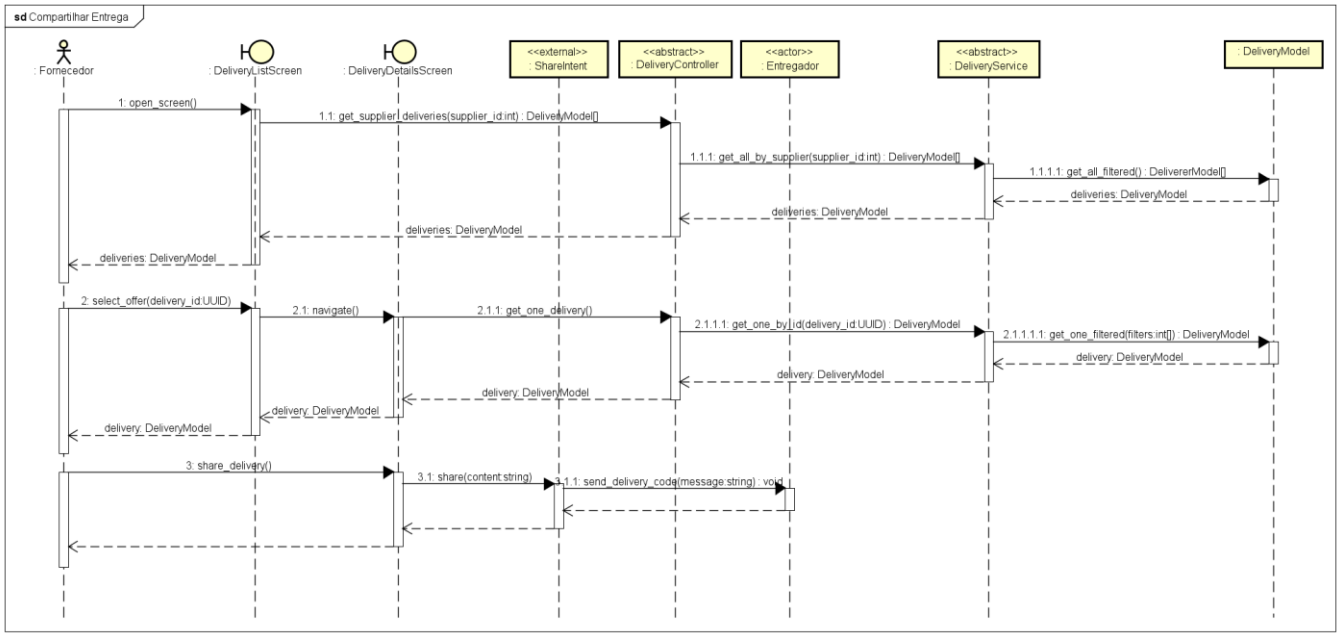


Figura 19. Diagrama de Sequência: Compartilhar entrega

A Figura 20 representa o fluxo de autenticação do entregador. Para execução deste processo, o entregador deve inserir o código de acesso à uma entrega recebida do fornecedor. Caso este código esteja correto, ele deve inserir seu celular para armazenamento. Este diagrama serve como detalhamento do diagrama representado pela Figura 6 e se relaciona com os casos de uso UC5 e UC6, por permitirem ao entregador acessar a aplicação ao mesmo tempo que solicitam a ele inserir seu número de telefone.

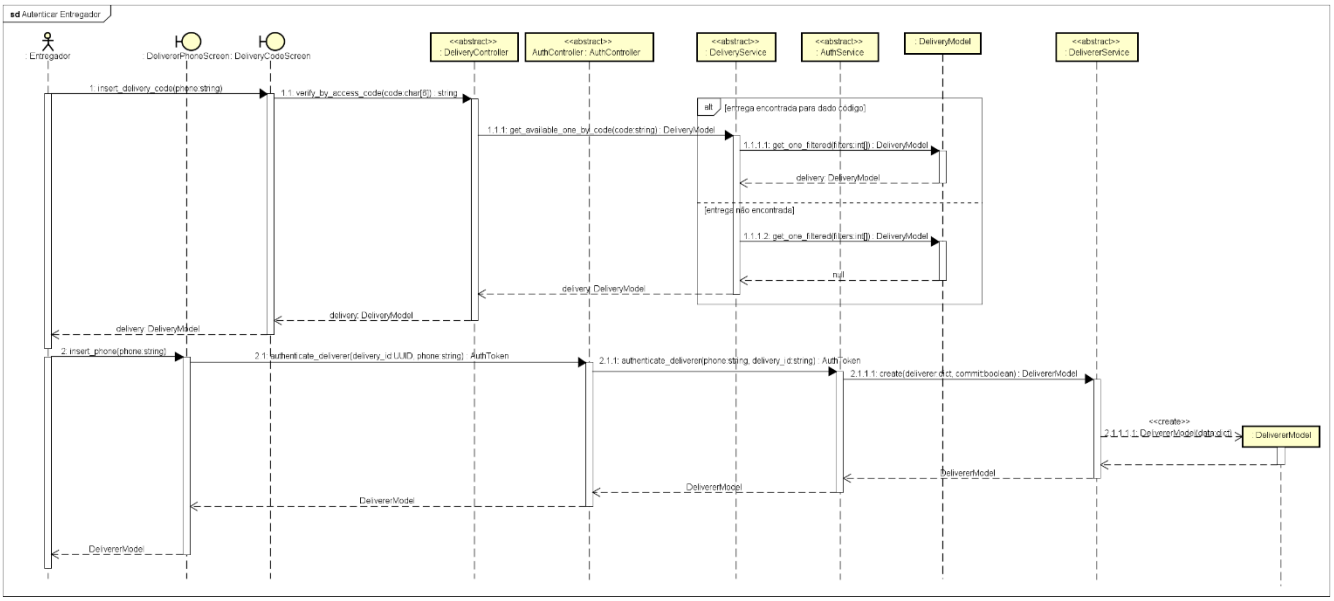


Figura 20. Diagrama de Sequência: Autenticar Entregador

A Figura 21 representa o fluxo de entrega de pedidos do sistema, esse que é realizado pelo entregador. Neste fluxo, o usuário deve, após selecionar a opção de iniciar entrega, repetir o processo de ver detalhes de um pedido e registrá-lo como finalizado ou como problema até que todos os pedidos acabem. Após esse processo ser finalizado, o sistema envia um evento informando a Ledger API de que a entrega foi finalizada. Este diagrama serve como detalhamento do diagrama representado pela Figura 7 e se relaciona com os casos de uso UC7, UC8, UC9, UC10, UC13, UC14, UC15, UC16 e UC17, representando o fluxo principal de interação com o sistema.

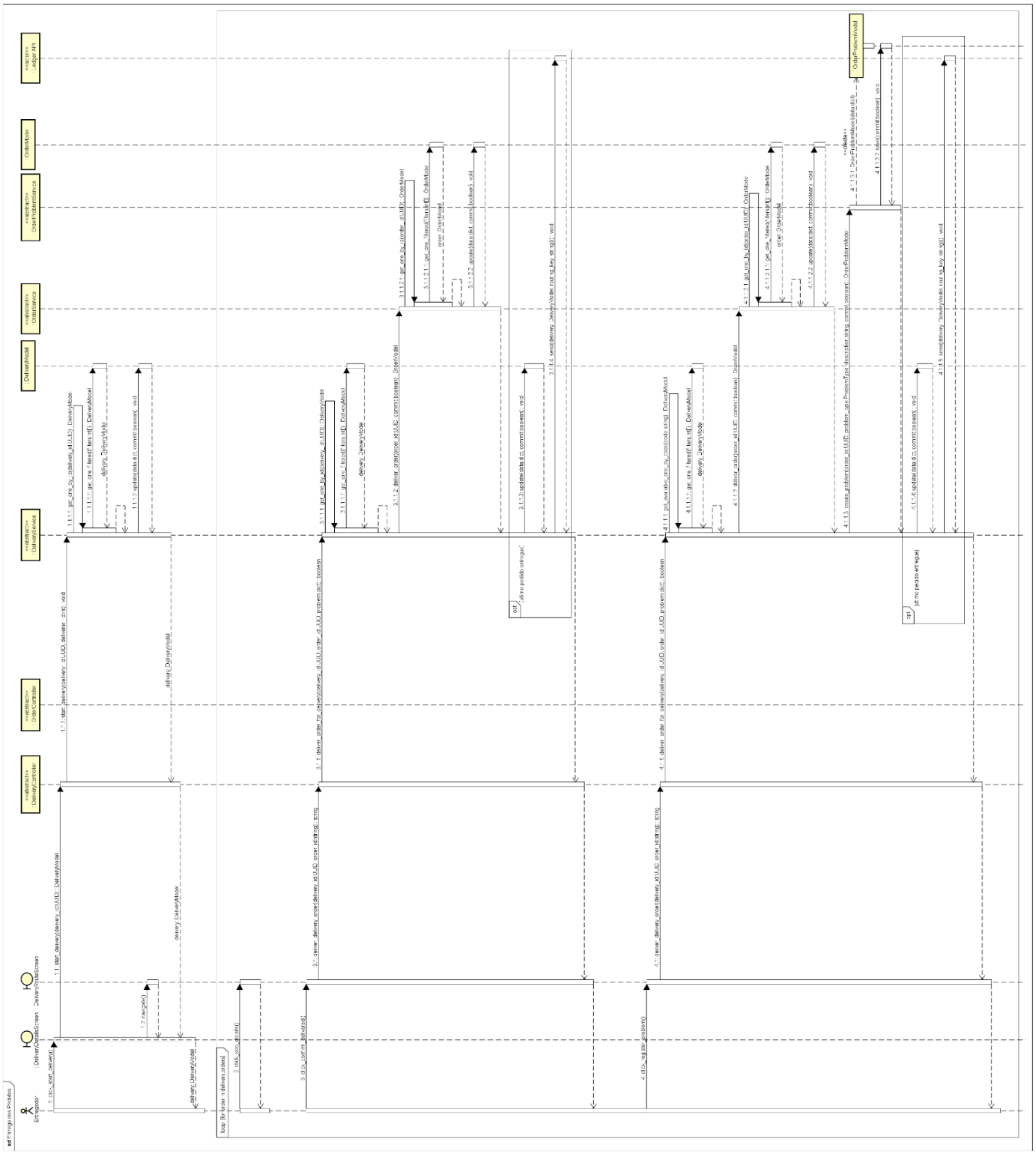


Figura 21. Diagrama de Sequência: Entrega dos pedidos

A Figura 22 representa o fluxo de envio do relatório diário para o gerente de operações. Esse fluxo se inicia por meio de da classe de DeliveryJobs, essa que é responsável por iniciar o processo a cada 24 horas. O processo consiste em recuperar todo as entregas ainda não incluídas em um relatório, e, caso exista pelo menos uma entrega, construir um relatório com elas, enviá-lo ao gerente de operações e, por fim, atualizá-las para que elas não sejam incluídas em um relatório futuro. Este diagrama serve como detalhamento do diagrama representado pela Figura 8 e se relaciona com o caso de uso UC19, que consiste no recebimento de um relatório diário de entregues por parte do ator citado.

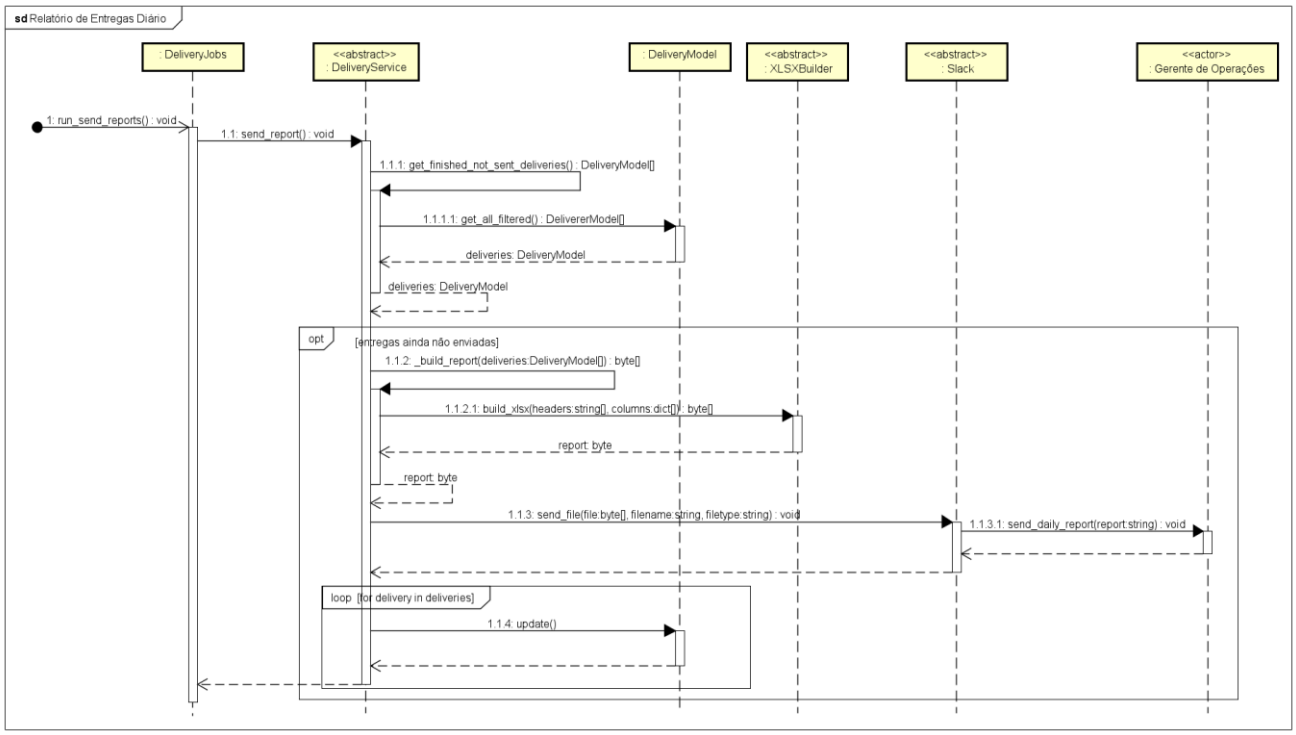


Figura 22. Diagrama de Sequência: Relatório de entregas diário

3.3 Diagramas de Comunicação

Nesta seção, são apresentados os diagramas de comunicação que modelam as trocas de mensagens dos diversos pacotes do sistema. O principal objetivo desses diagramas é representar todos os agentes e as mensagens trocadas entre eles no sistema. Por conta disso, os principais fluxos da aplicação são representados como forma de documentar as comunicações que os compõem.

A Figura 23 representa as trocas de mensagens ocorridas durante o processo de gerar uma rota otimizada. Esse processo se inicia por meio de uma mensagem recebida informando que uma oferta foi concluída, gerando assim uma entrega programada e uma rota de entrega otimizada para ela. Este fluxo se relaciona com o diagrama representado pela Figura 17. Ele também se relaciona aos casos de uso UC11, UC12, UC3 e UC10, gerando a rota de entrega que será visualizada por ambos fornecedores e entregadores.

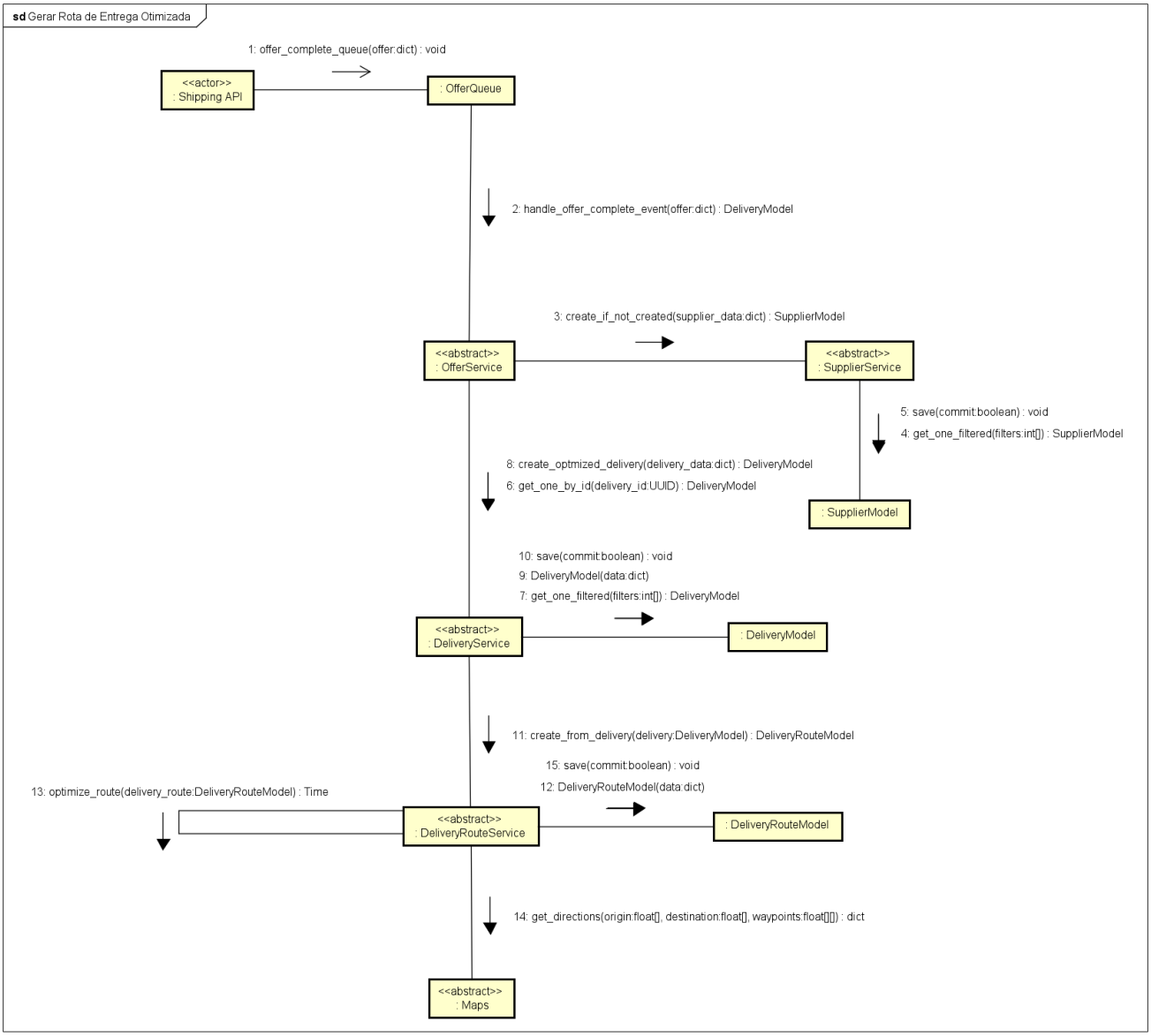


Figura 23. Diagrama de Comunicação: Gerar rota de entrega otimizada

A Figura 24 representa as trocas de mensagens existentes no fluxo de autenticação do fornecedor. Neste fluxo as mensagens mais relevantes são as trocas com a Auth API, serviço externo de autenticação dos usuários. Este diagrama se relaciona diretamente com o diagrama de sequência representado pela Figura 18. Dessa mesma forma, ele também se relaciona aos casos de uso UC1 e UC2, por representar o processo de entrada do fornecedor na aplicação.

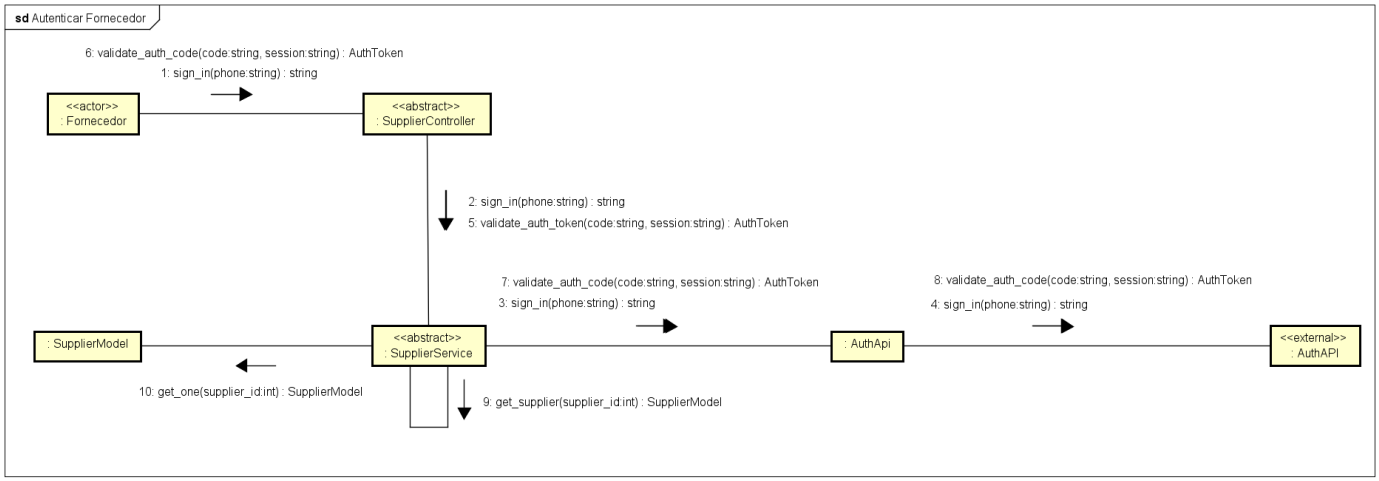


Figura 24. Diagrama de Comunicação: Autenticar Fornecedor

A Figura 25 representa as trocas de mensagens referentes ao processo de compartilhar uma entrega com um entregador. Dentro deste fluxo, é gerada uma mensagem a ser compartilhada com um entregador por meio da API de compartilhamento interna do aparelho do usuário, representada por meio do ShareIntent. Essa troca de mensagens se relaciona com o diagrama da Figura 19, assim como o caso de uso UC4, em que o fornecedor deve ser capaz de compartilhar a entrega com um entregador.

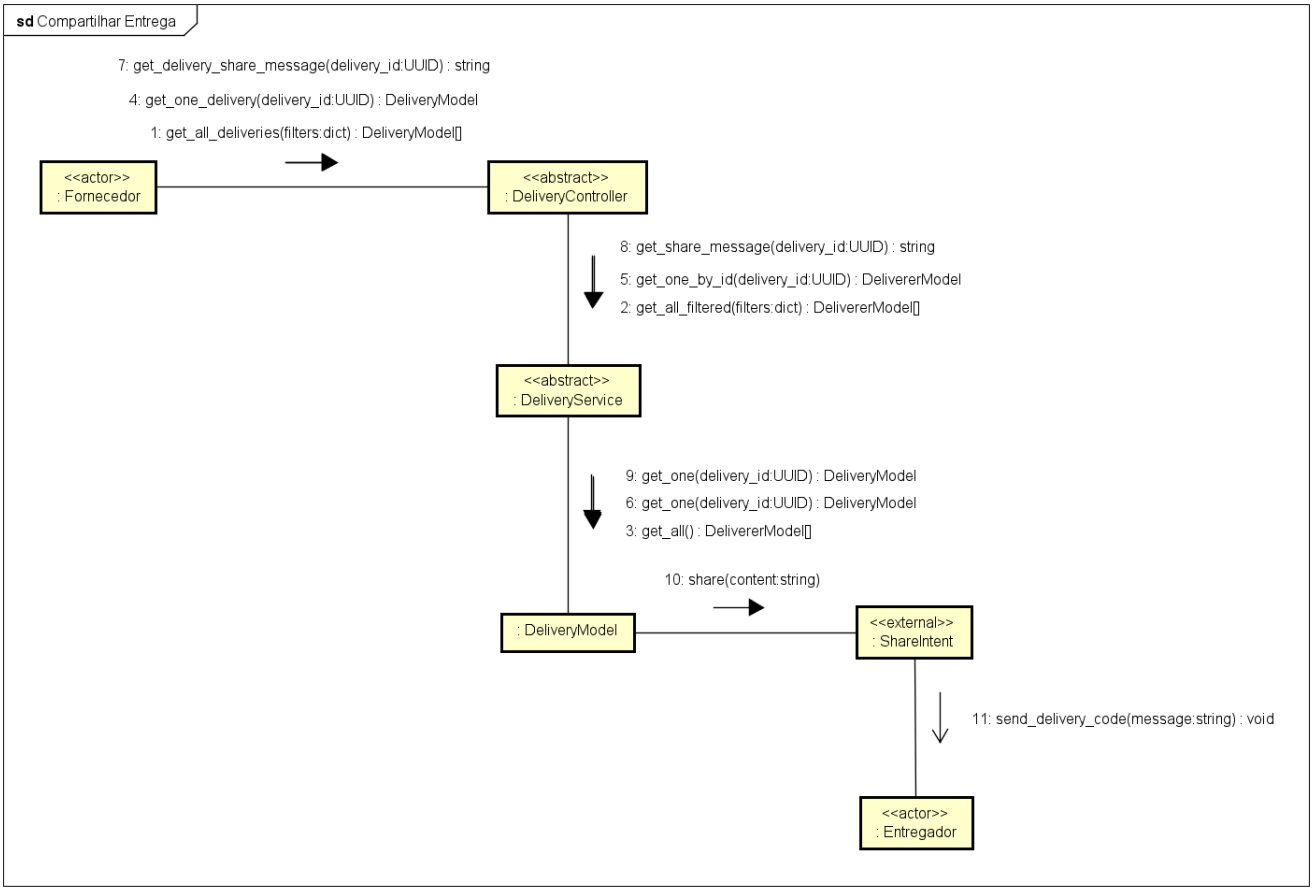


Figura 25. Diagrama de Comunicação: Compartilhar entrega

A Figura 26 representa as trocas de mensagens que ocorrem durante o fluxo de autenticação do entregador. Durante esse fluxo, também acontece o registro do telefone do entregador na base de dados. Esta modelagem se relaciona com a representada pela Figura 20, assim como os casos de uso UC5 e UC6, que solicitam que o entregador seja capaz de acessar a aplicação ao mesmo tempo que solicita a ele inserir seu telefone.

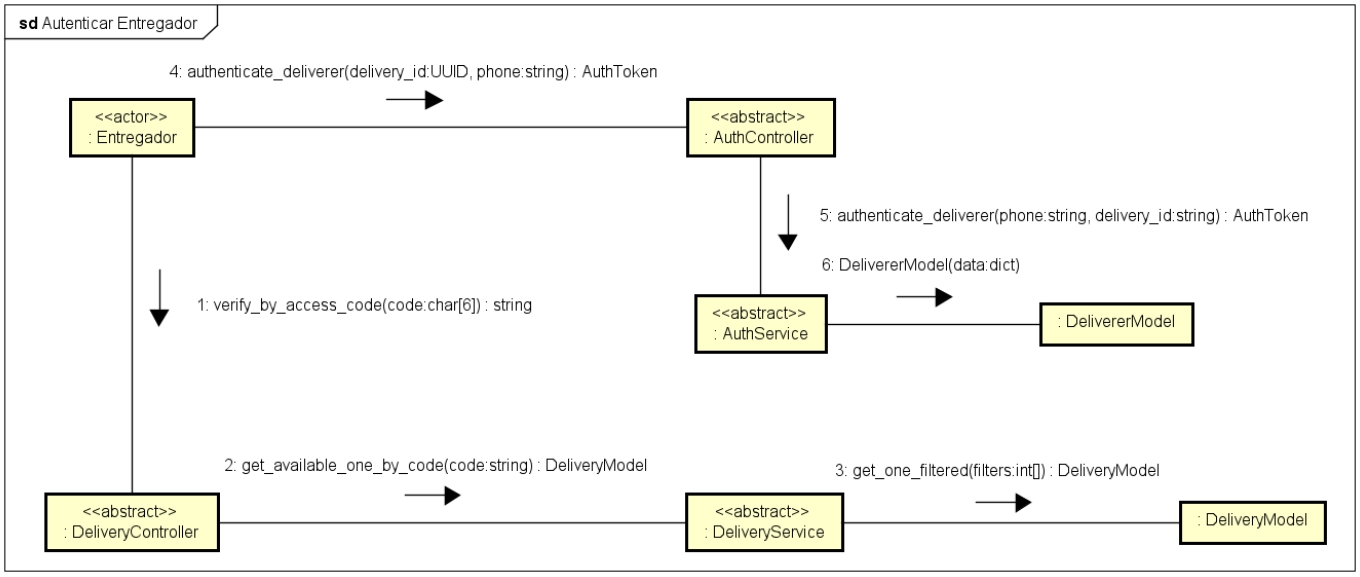


Figura 26. Diagrama de Comunicação: Autenticar Entregador

A Figura 27 representa a troca de mensagens existentes no processo de entrega dos pedidos de uma oferta. Esse é o fluxo principal da aplicação, partindo da tela de visualização dos detalhes de uma entrega, passando pela entrega de cada produto individual, registrando possíveis problemas, e finalizando ao entregar o último produto, enviando um evento de entrega finalizada. Este diagrama serve como detalhamento do diagrama representado pela Figura 7. Dessa mesma forma, ele se relaciona com os casos de uso UC7, UC8, UC9, UC10, UC13, UC14, UC15, UC16 e UC17, representando o fluxo principal de interação com o sistema.

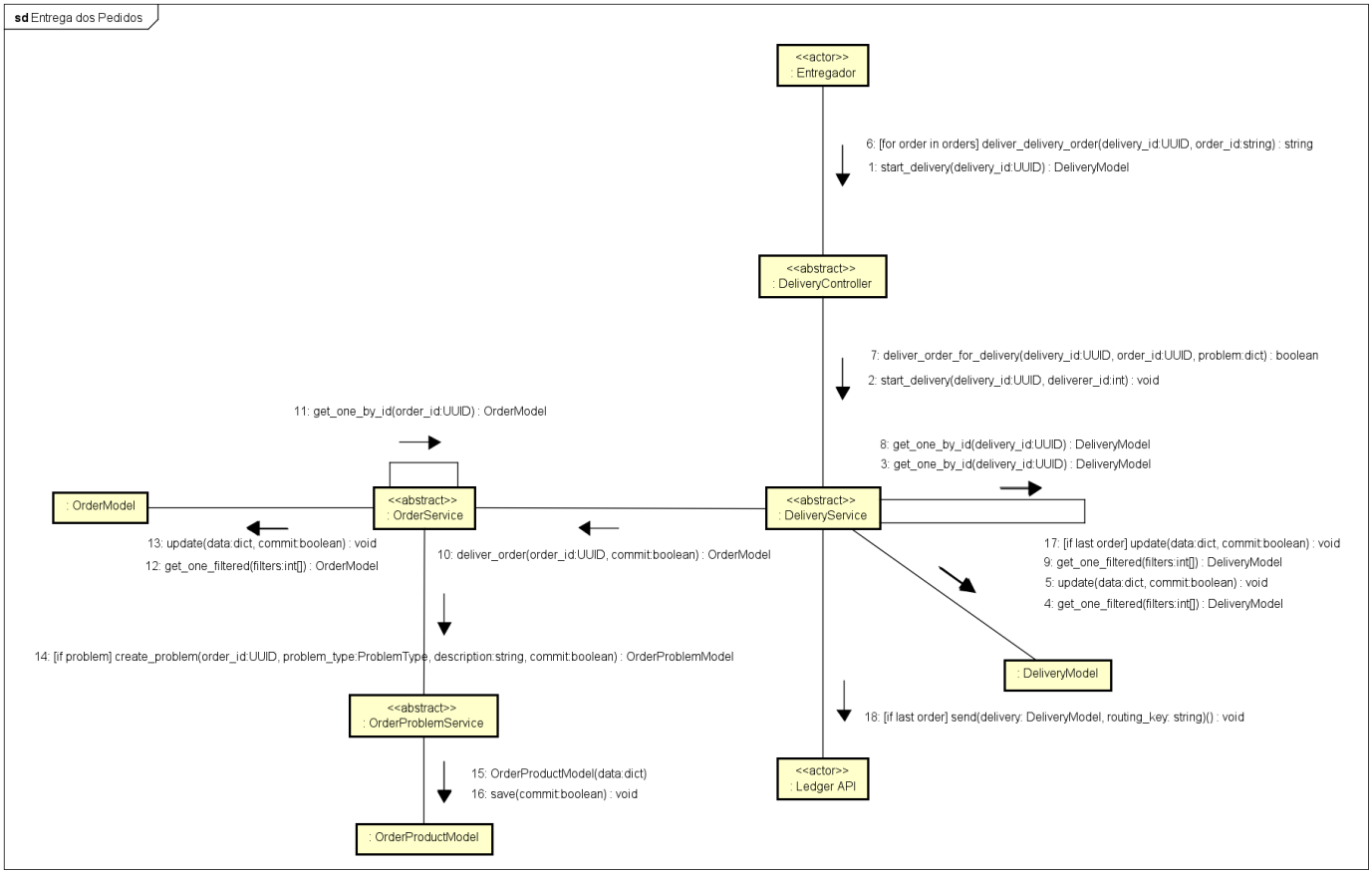


Figura 27. Diagrama de Comunicação: Entrega dos Pedidos

A Figura 28 representa a troca de mensagens que ocorre durante o processo de envio do relatório diário ao gerente de operações. É relevante mencionar que esse processo ocorre de forma periódica, sendo ativado pela DeliveryJobs a cada 23 horas. Ele se relaciona com o diagrama de sequência representado pela Figura 22. Assim como com o caso de uso UC19, que consiste no recebimento de um relatório diário de entregues por parte do ator citado.

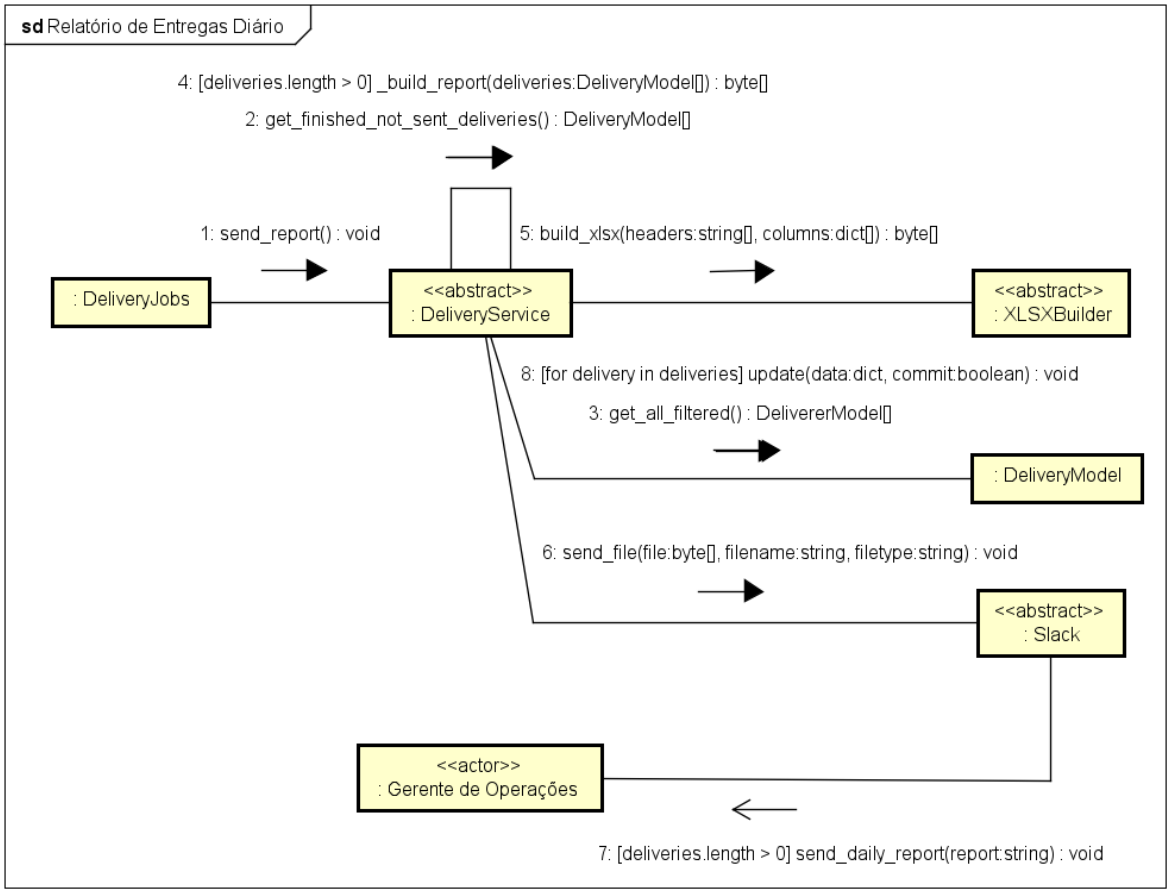


Figura 28. Diagrama de Comunicação: Relatório de entregas diário

3.4 Arquitetura

A Figura 29 representa o diagrama de pacotes, ou de arquitetura, do sistema. Nele é possível observar três diferentes sistemas representados. O primeiro é da aplicação *mobile* que contém a Interface de Usuário (UI, do inglês *User Interface*) do sistema, esse que depende das rotas disponibilizadas pelo pacote DeliveryAPI. Esse por sua vez contém dois pacotes, responsáveis pela lógica de negócios, *bussiness,* e responsável pela lógica de dados, *data.*

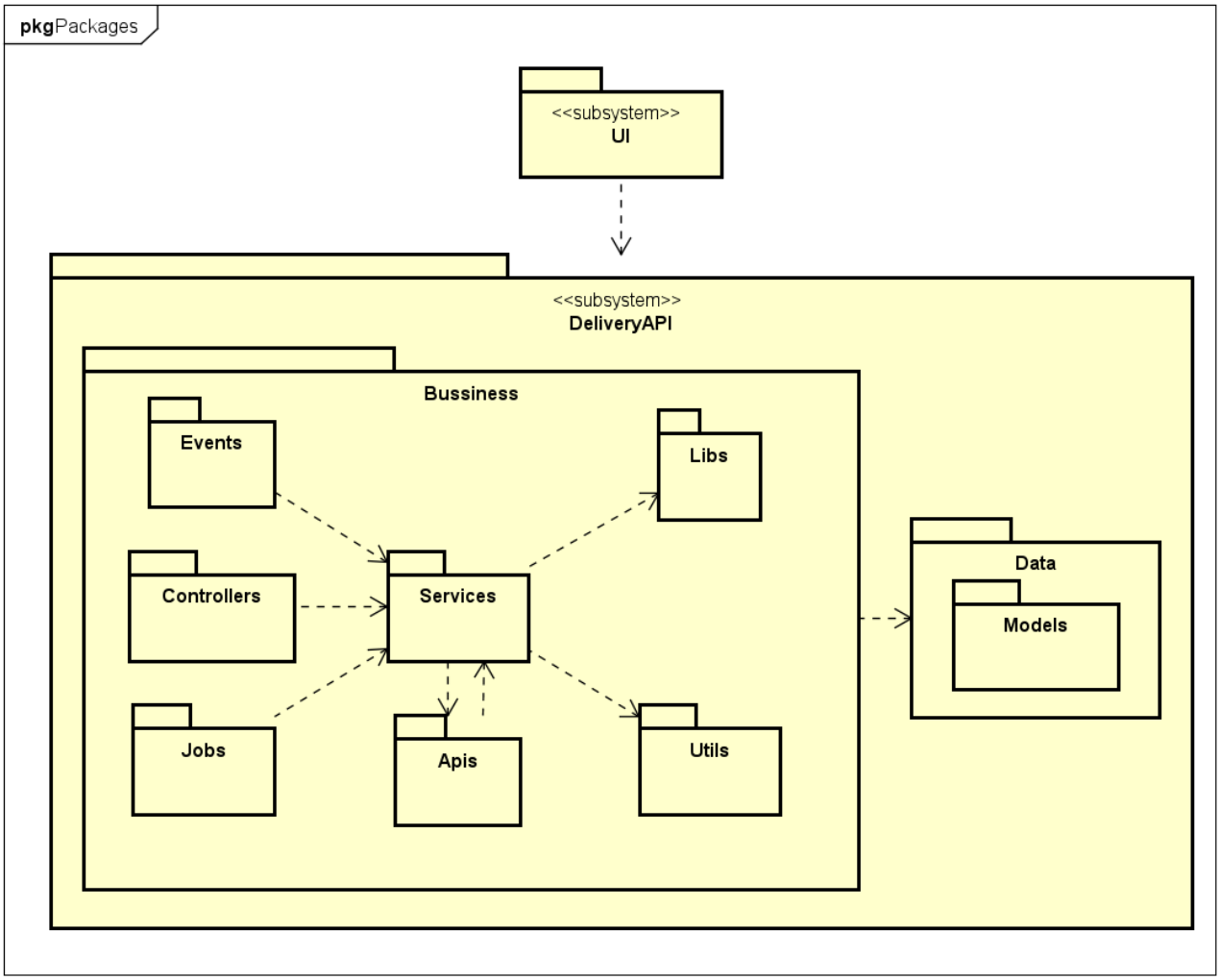


Figura 29. Diagrama de Pacotes da Aplicação

3.5 Diagramas de Estados

O diagrama representado pela Figura 30 contém a lógica de mudança de estados pelos quais uma entrega passa. Nele é possível perceber que a entrega possui três estados: criada, em progresso e finalizada. O primeiro estado se dá assim que a oferta é criada no banco de dados, sendo o estado padrão do objeto tratado. O segundo estado é o de em progresso, durante esse estado cada pedido da entrega deve ser registrado como entregue ou como problema, sendo que após a entrega do último pedido, a entrega tem seu estado modificado para finalizada. Em seu terceiro estado, a entrega tem o tempo necessário para ser concluída registrado e, dessa mesma forma, uma mensagem é enviada notificando as demais aplicações do sistema de que a entrega foi finalizada.

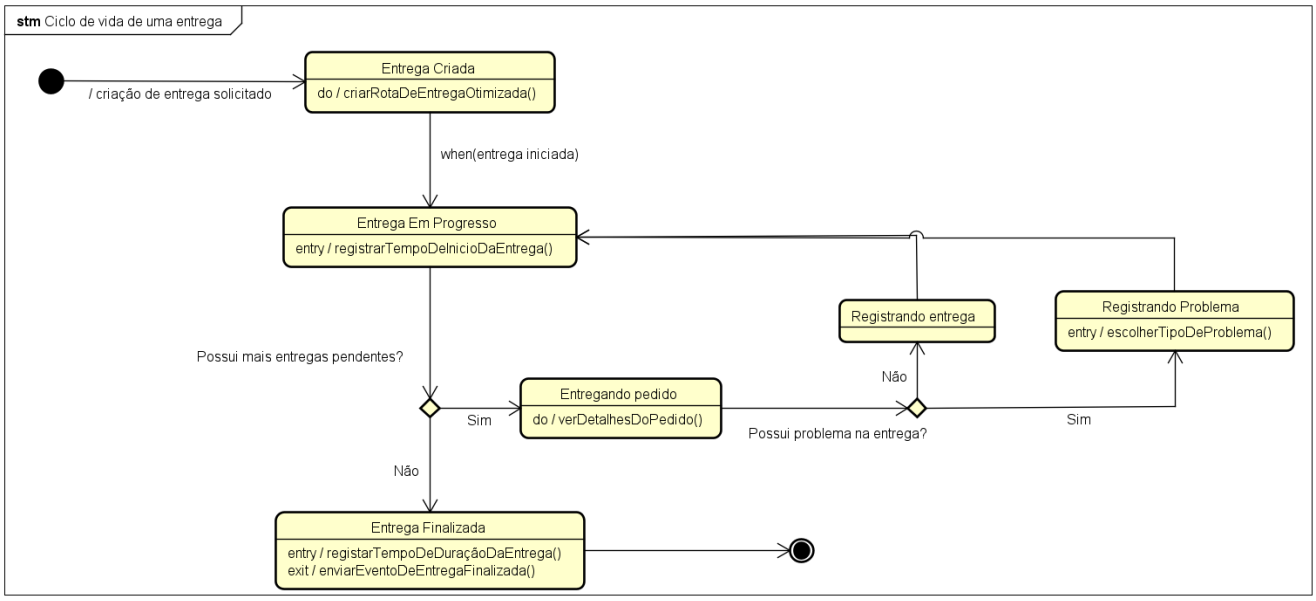


Figura 30. Diagrama de estados: Ciclo de vida de uma entrega

3.6 Diagrama de Componentes e Implantação.

A Figura 31 representa o diagrama de componentes do sistema desenvolvido. Nele é possível ver 5 componentes do projeto proposto: Mobile APP, Delivery API, Message Queue, SGBD e Zapt APIs. O componente Mobile APP representa a aplicação móvel a ser desenvolvida. Ele é dividido em duas partes, uma contendo a interface do usuário, essa que possui as páginas, componentes e estados necessários para tornar a aplicação utilizável, e os *services*, esses que permite à aplicação se comunicar com o servidor representado pela Delivery API por meio de rotas HTTP. Portanto, a Delivery API contém os diversos pacotes necessários para o funcionamento da aplicação. Dentro deste componente os dados devem surgir por meio de um controlador, *job* ou evento, ser enviado à um *service*, esse que contém a lógica de negócios da aplicação, e se comunicará com um modelo de dados, API externa, biblioteca ou classe utilitária. O componente Message Queue representa o sistema de mensageria existente entre a aplicação desenvolvida e as aplicações externas existentes. Ele interage diretamente com o pacote de eventos da Delivery API, permitindo envio e recebimento de mensagens. Esse mesmo pacote interage com a Ledger API e Shipping API, serviços externos que, respectivamente, recebem e enviam mensagens. O SGBD representa o banco de dados utilizado para armazenar os dados da aplicação, ele se comunica com o pacote de modelos da Delivery API por meio da linguagem SQL.

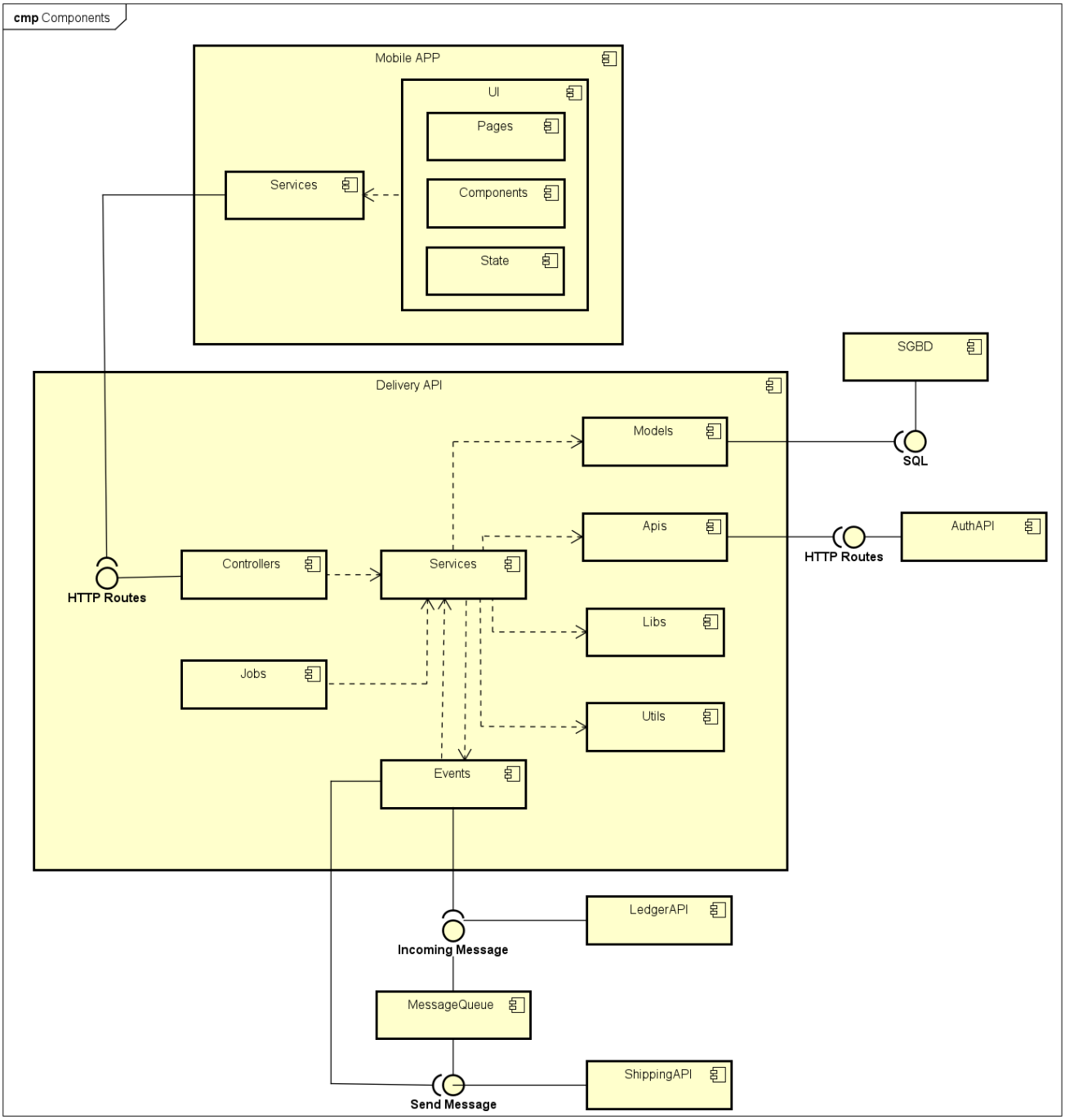


Figura 31. Diagrama de Componentes da Aplicação

A Figura 32 representa o diagrama de implantação da aplicação descrita neste documento. Nele é possível ver 5 principais nós. O primeiro nó, o Dispositivo Android/IOS, representa o um dispositivo móvel, esse que executará a aplicação móvel contendo a UI da aplicação. O nó Servidor de Comunicação permitirá a comunicação entre os nós Dispositivo

Android/IOS e Servidor da Aplicação por meio de um balanceador de carga. O nó Servidor da Aplicação contém o container que executa a Delivery API, aplicação aqui documentada. Esse nó deve ser comunicar com os dois últimos nós representados, o Agende de Mensagens, que permitirá a comunicação por meio de um sistema de mensageria, e o SGBD, que permitirá armazenamento de dados por meio de um banco de dados SQL.

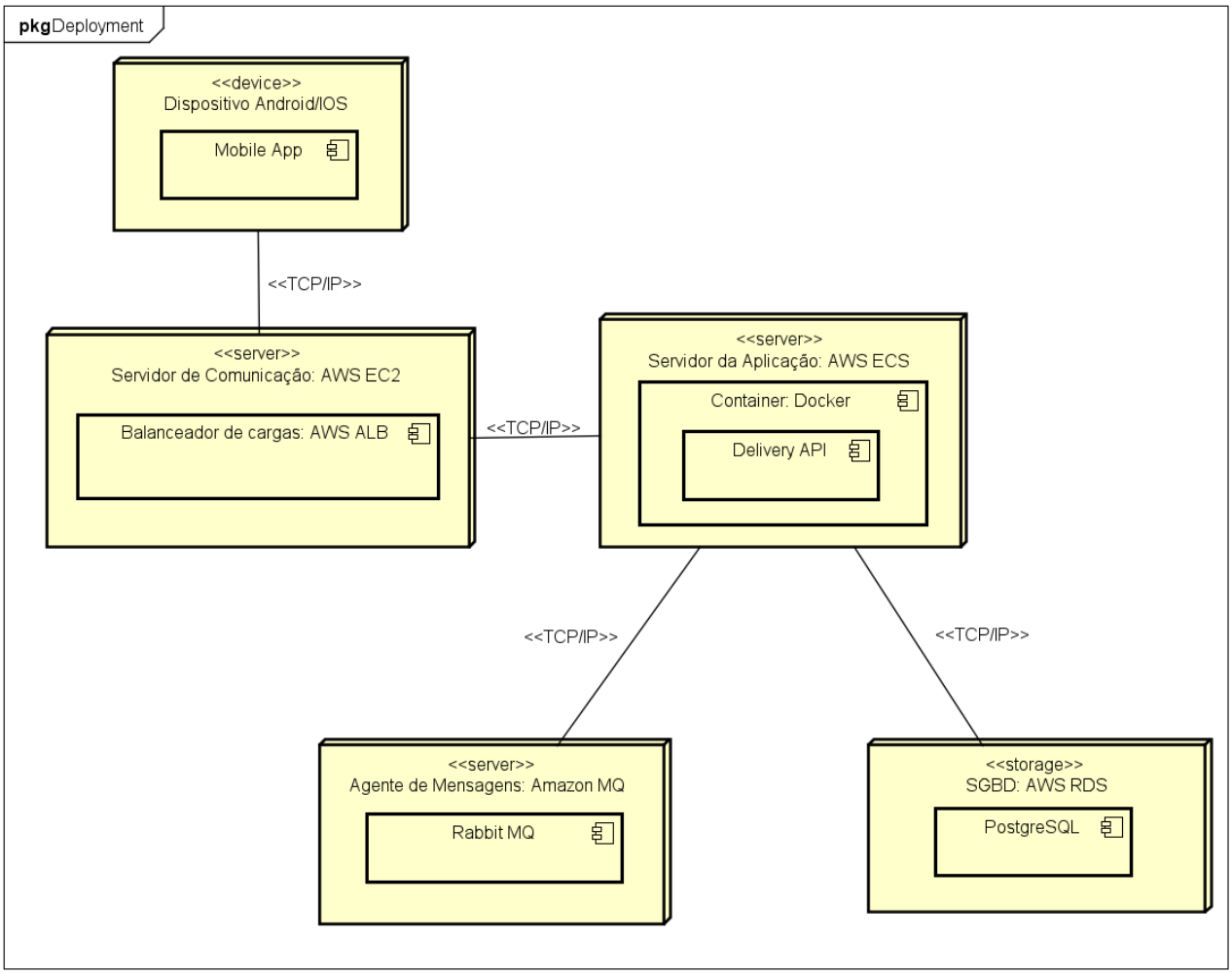


Figura 32. Diagrama de implantação da Aplicação

4. Projeto de Interface com Usuário

Está seção tem como objetivo mostrar e descrever as interfaces de interação com o usuário das quais a aplicação é composta. Para isso foi realizado um *mockup* de alta fidelidade usando a ferramenta Figma. Dessa mesma forma, as interfaces foram relacionadas com os

casos de usos especificados na Seção 2.3.1 a fim de mapear todas as funcionalidades necessárias para cumprimento dos requisitos especificados.

4.1 Esboço das Interfaces Comuns a Todos os Atores

A Figura 33 representa a página de *Splash Screen* da aplicação. Esta página deve ser mostrada quando for aberta e enquanto algum processamento em plano de fundo estiver ocorrendo. Um exemplo de processamento seria a autenticação de um usuário que já se autenticou na aplicação. Esta tela redireciona o usuário automaticamente à página representada pela Figura 35.

Figura 33. Splash Screen

Notificações gerais da aplicação devem ser feitas por meio do uso do componente *snackbar.* A Figura 34 representa um exemplo do uso desse componente, para uma notificação de sucesso, mais especificamente da confirmação da entrega de um pedido. Este componente pode ter sua cor modificada conforme necessário, usando tons de amarelo ou vermelho para representar, por exemplo, mensagens de atenção ou erro.

Figura 34. Notificação de sucesso

A Figura 35 representa o fluxo de preenchimento da página de inserção do código de entrega. Nela o entregador deve inserir o código de compartilhamento de entrega recebido do fornecedor sendo redirecionado para a Figura 41. Caso o fornecedor esteja acessando

a aplicação, ele deve selecionar a opção “Sou um parceiro Zapt” e seguir o fluxo de autenticação por telefone, iniciado na Figura 36. Essa tela representa o ponto de entrada da aplicação, estando conectada com o UC1 e UC5, ambos casos de uso que permitem ao Entregador e Fornecedores acessarem a aplicação.

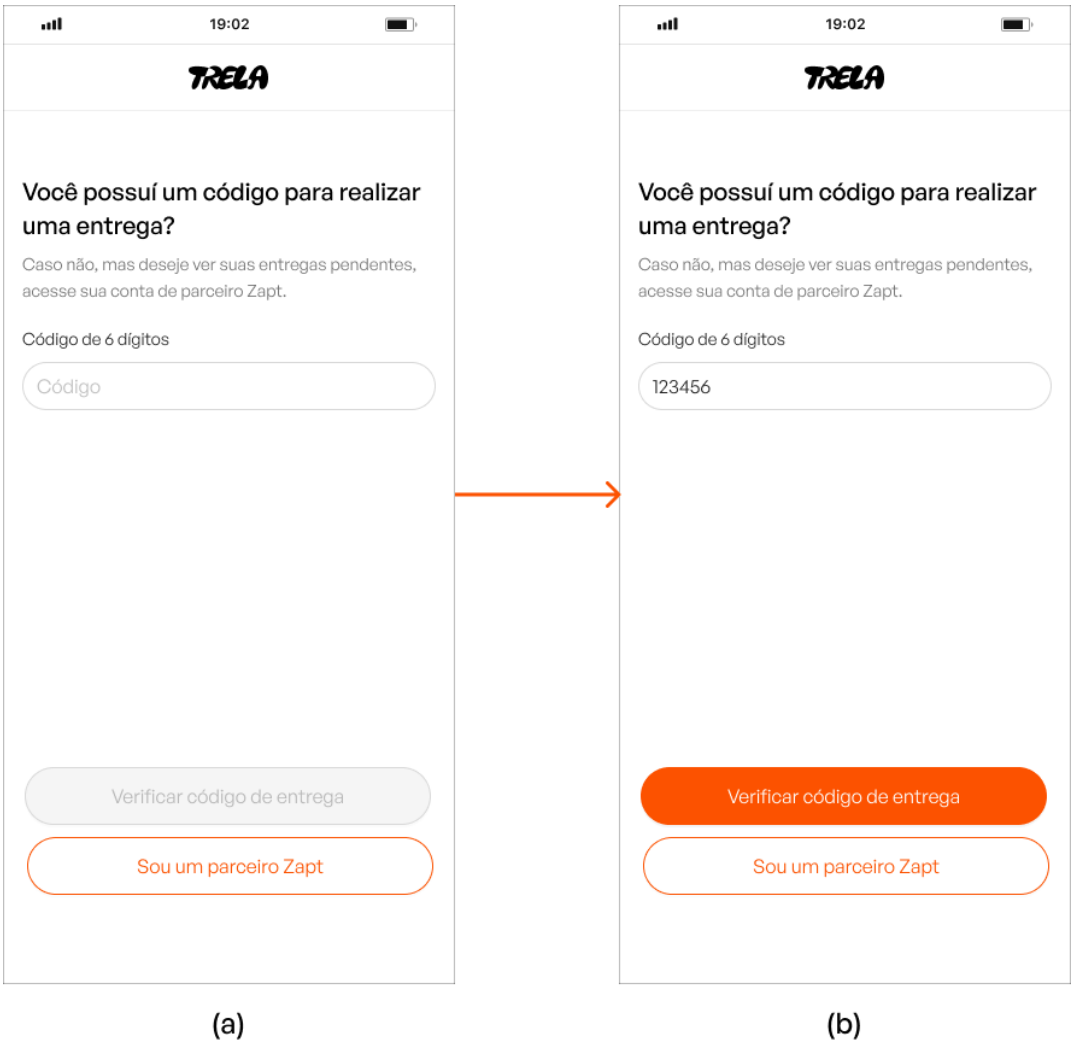


Figura 35. Página de inserção do código de offerta vazia (a) e preenchida (b)

4.2 Esboço das Interfaces Usadas pelo Fornecedor

A Figura 36 representa a primeira parte do fluxo de autenticação específico ao fornecedor. Nela é pedido o telefone do usuário, esse que está conectado à sua conta previamente cadastrada. Desta forma, cada Subfigura representa um estado de preenchimento do

formulário, sendo eles o estado vazio, com telefone preenchido e com erro no campo de telefone. Esta Figura se relaciona com os casos de uso UC1 e UC2, sendo parte do processo de autenticação do fornecedor. Após preenchimento do telefone de forma correta, o usuário é redirecionado à página representada pela Figura 37.

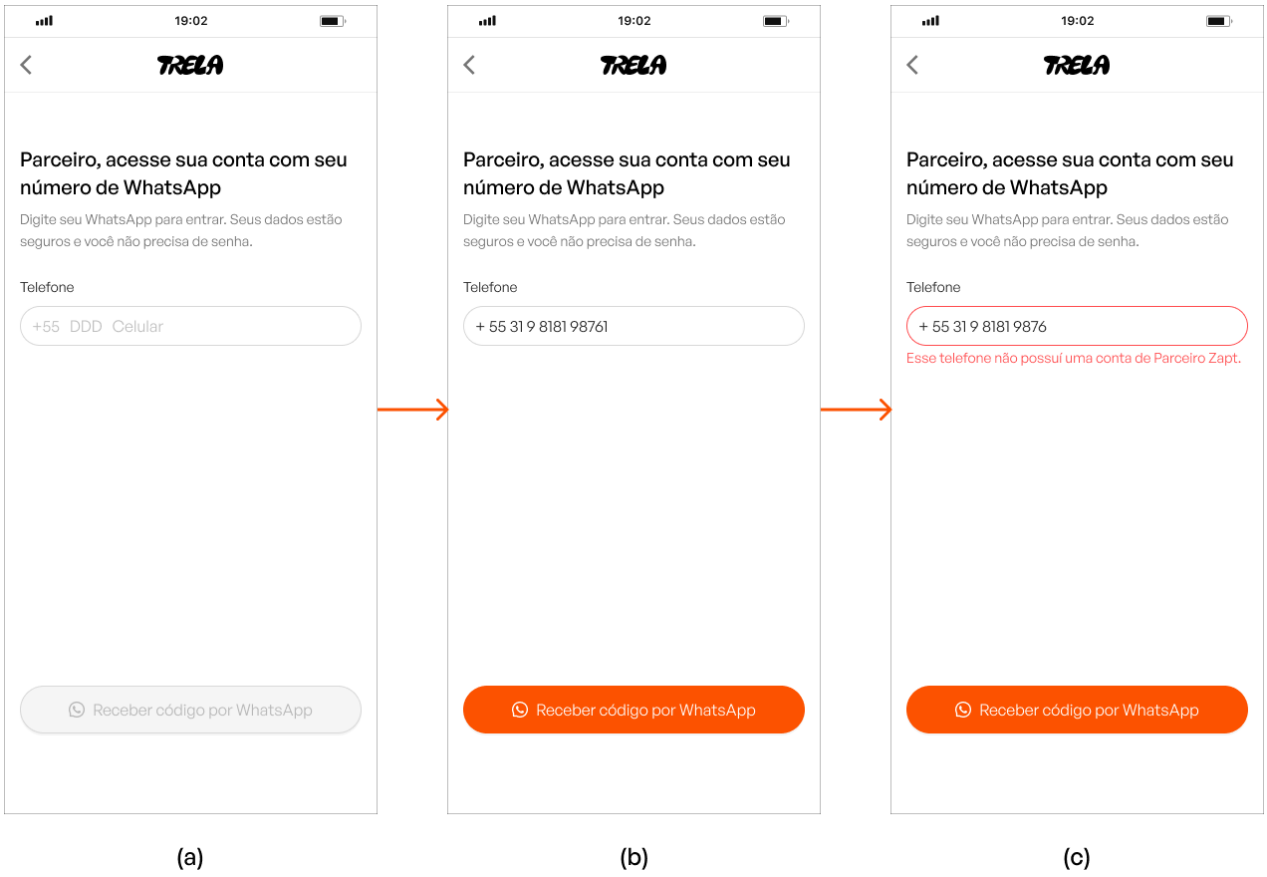


Figura 36. Página do telefone do fornecedor vazia (a), preenchida (b) e com erro (c)

Na Figura 37 é possível ver a página de inserção do código de validação enviado ao Whatsapp do telefone preenchido previamente no fluxo de entrada do fornecedor. Após preenchimento do código, a página muda de estado de acordo com a Figura 37 (b). Caso o código seja validado com sucesso, o usuário é enviado à página da Figura 38. Esta página está relacionada aos casos de uso UC1 e UC2, fazendo parte do fluxo de autenticação dos fornecedores.

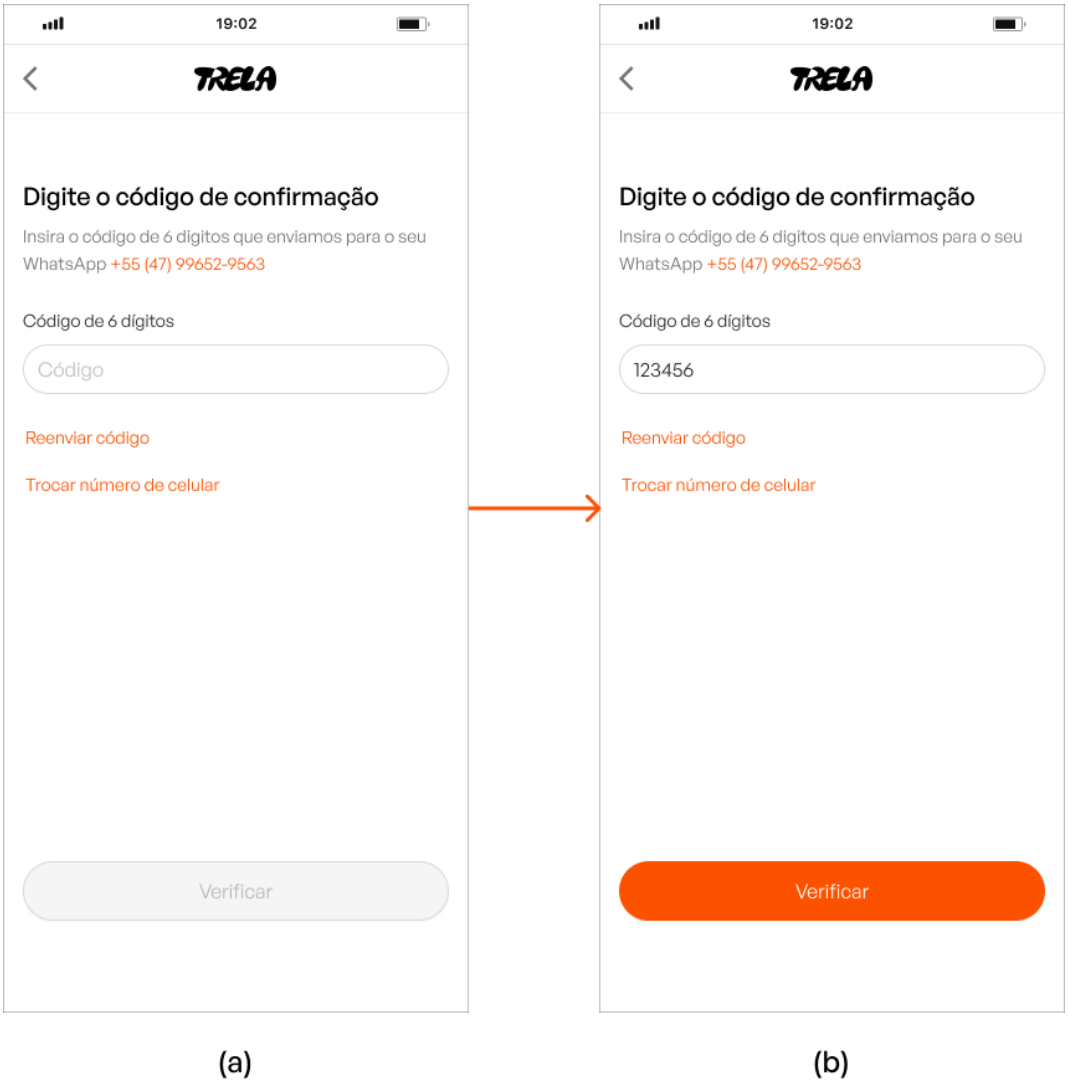


Figura 37. Página de inserção do código de validação, vazio (a) e preenchido (b)

A Figura 38 representa o conteúdo das três abas da tela contendo as listas das ofertas. Cada aba contém, respectivamente, uma lista de ofertas que, ainda devem ser entregues (Figura 38 (a)), estão sendo entregues neste momento (Figura 38 (b)) e que já foram entregues (Figura 38 (c)). A partir destas telas também é possível acessar, o perfil da conta do usuário conectado, representado na Figura 40, e a página de detalhes da oferta, detalhada na Figura 39, servindo de página central para o fornecedor dentro da aplicação. Estas páginas se relacionam com o caso de uso UC3, por mostrarem ao fornecedor as listas de ofertas que devem ser entregues.

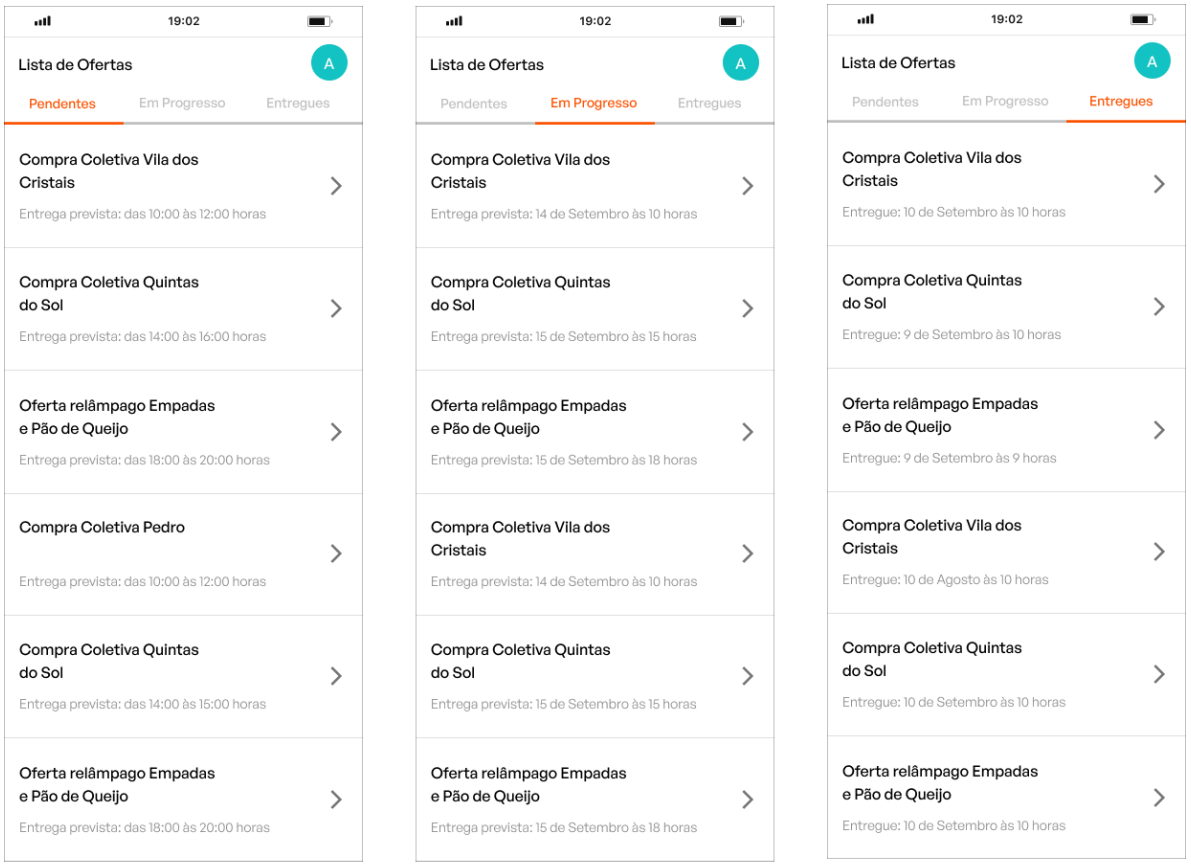


Figura 38. Lista de ofertas com entrega do dia

A Figura 39 representa a página com os detalhes da oferta previamente selecionada. Nela é possível ver detalhes como a data da entrega e localização no mapa. Dessa mesma forma é possível ver uma lista de produtos ou de pedidos contidos na entrega. Nesta página também é possível enviar a oferta à um entregador para que ele a realize. Com base nos detalhes da imagem, é possível relacioná-la ao caso de uso UC4, no qual o fornecedor deve ser capaz de compartilhar uma entrega com um entregador.

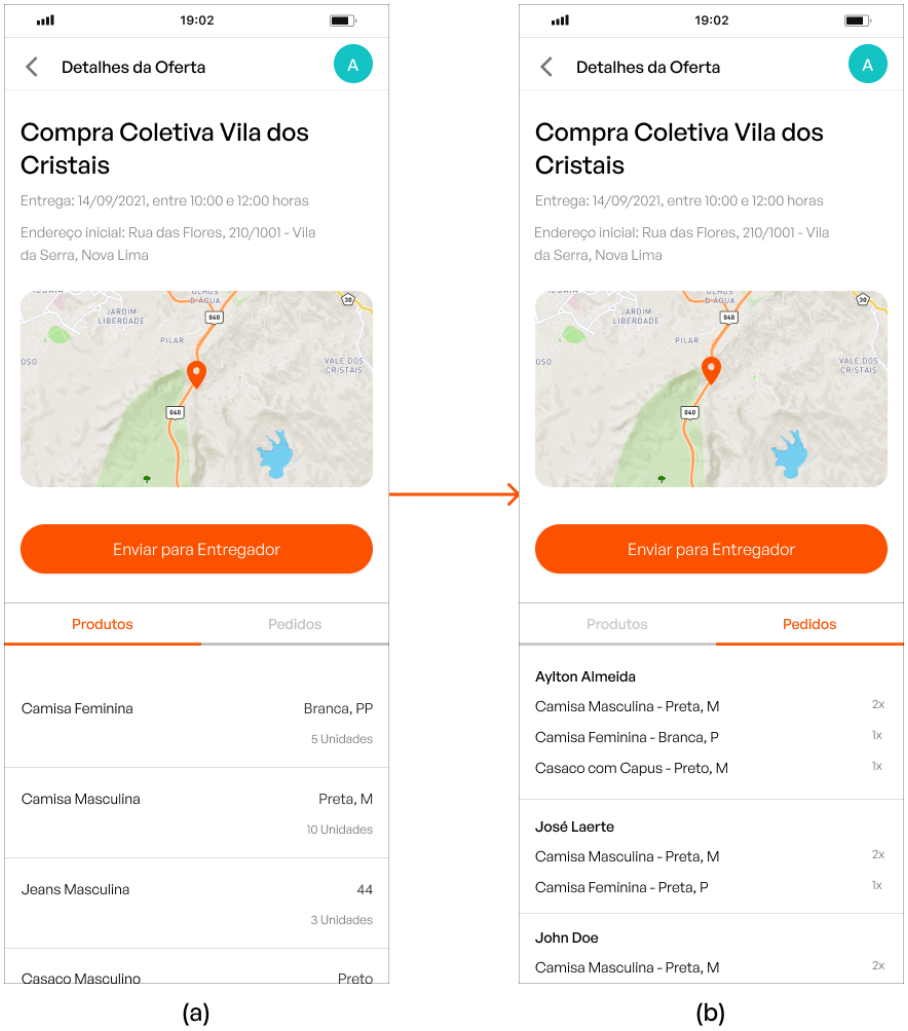


Figura 39. Página de detalhe da Oferta

A Figura 40 representa a página contendo o perfil do usuário conectado. Nela é possível ver os dados básicos do fornecedor, como nome e CNPJ. Também é possível por meio desta tela sair da aplicação ou voltar para a página anterior. Está página se relaciona com o caso de uso UC20, no qual é especificado a necessidade do fornecedor de sair da aplicação caso desejado.

Figura 40. Página com Perfil do Usuário

4.3 Esboço das Interfaces Usadas pelo Entregador

A Figura 41 mostra a tela de inserção do telefone do entregador. Ela se assemelha à página representada pela Figura 36, possuindo apenas mensagens diferentes. Ao ser preenchida, ela adquire o estado representado pela Figura 41 (b). Após preenchimento do telefone, o usuário é redirecionado à página contendo os detalhes da entrega a ser realizada, representada pela Figura 42. Esta tela está relacionada ao caso de uso UC6, por pedir a inserção de seu telefone ao entregador.

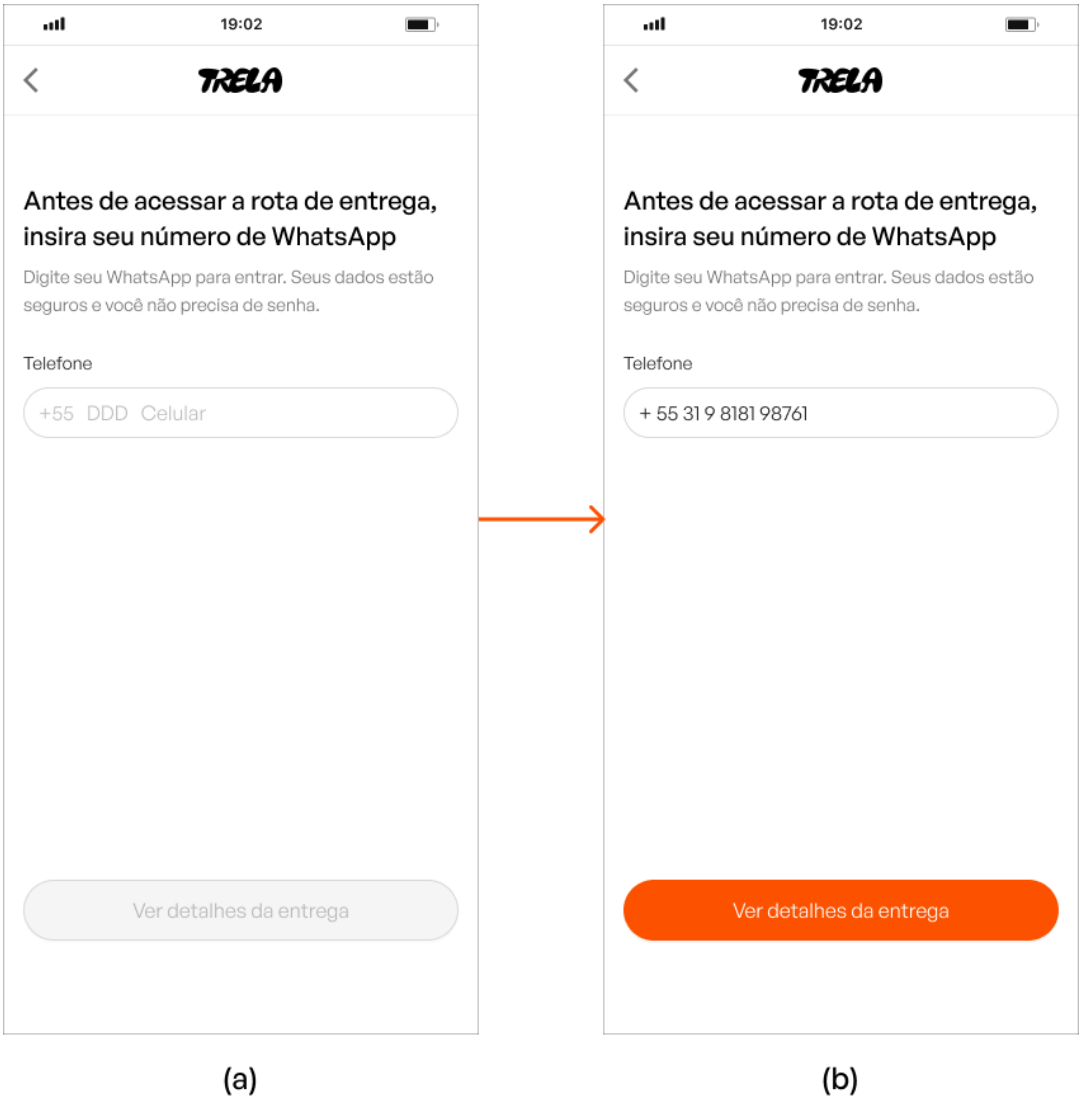


Figura 41. Página do telefone do entregador vazia (a) e preenchida (b)

A Figura 42 representa a página de detalhes da entrega. A partir dela o entregador consegue ver quais produtos ele precisa levar, rota da entrega e tempo estimado para realizá-la. Além disso, ele é capaz de cancelar o processo de entrega ou iniciá-la, ambas ações necessitam a confirmação do usuário por meio de um modal, representados respectivamente pela Figura 42 (a) e Figura 42 (b). As páginas mencionadas se relacionam aos casos de uso UC7, permitindo ao entregador iniciar o processo de entrega, e UC10, mostrando a rota completa de entrega ao usuário.

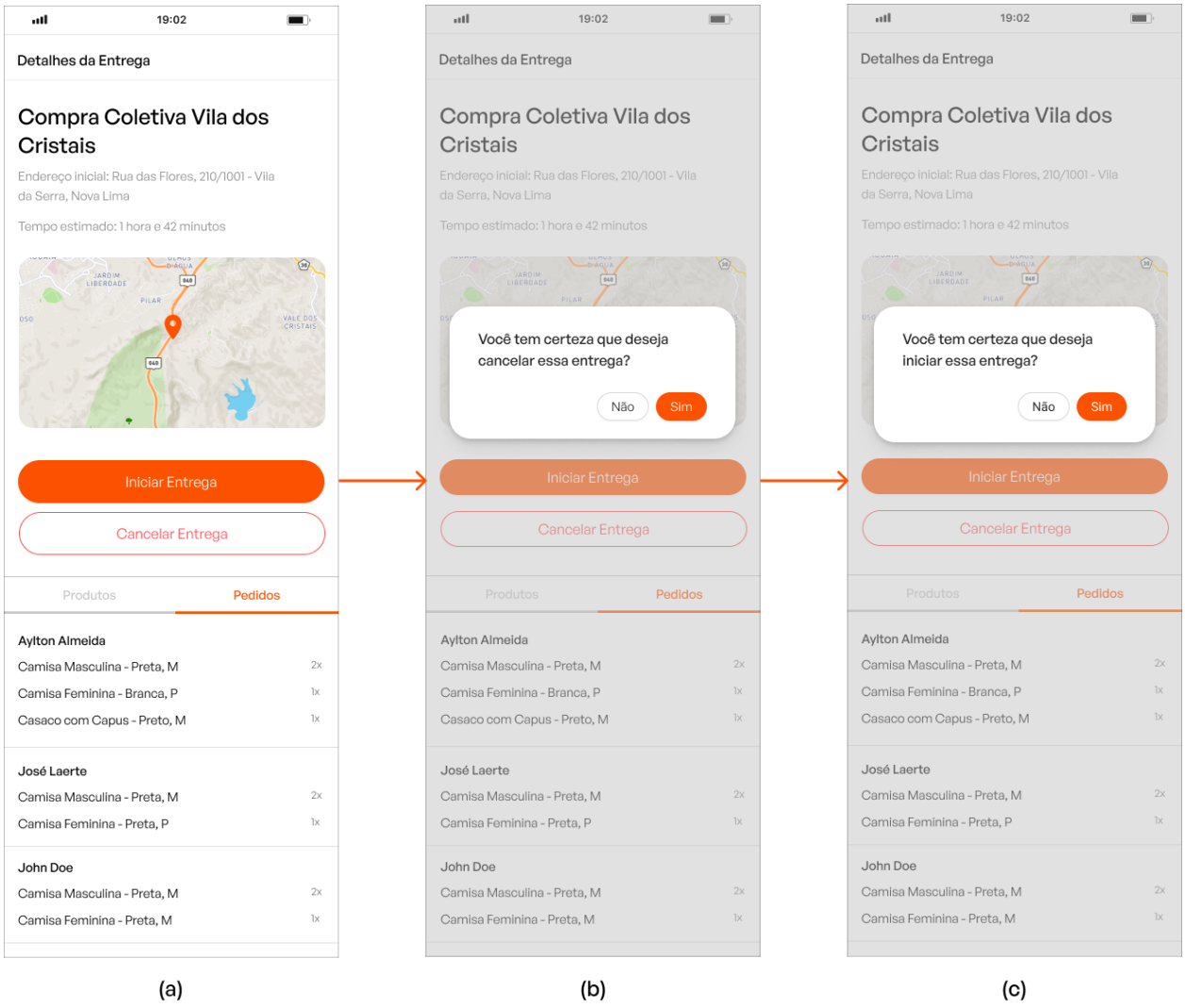


Figura 42. Página de detalhes da entrega padrão (a), com modal de cancelamento (b) e com modal de confirmação do início da entrega (c)

A Figura 43 contém a página com a rota de entrega para o próximo pedido. O fluxo sugere que à medida que os pedidos são entregues, o próximo é liberado para que o entregador o visualize. Esta tela possui dois estados, o primeiro, representado pela Figura 43 (a) é o de detalhes minimizados, em que apenas nome do próximo destinatário e horário previsto de entrega são exibidos. Já o segundo estado da tela, representado pela Figura 43 (b) permite ao usuário visualizar mais dados da entrega, como horário esperado de término e endereço, além do tempo até o próximo destino e o nome do destinatário. Por fim, também existe um botão para visualização dos detalhes do pedido, que redireciona o usuário à Figura 45. Essa página se relaciona com o caso de uso UC8, permitindo ao usuário visualizar a rota do próximo destino de entrega previsto.

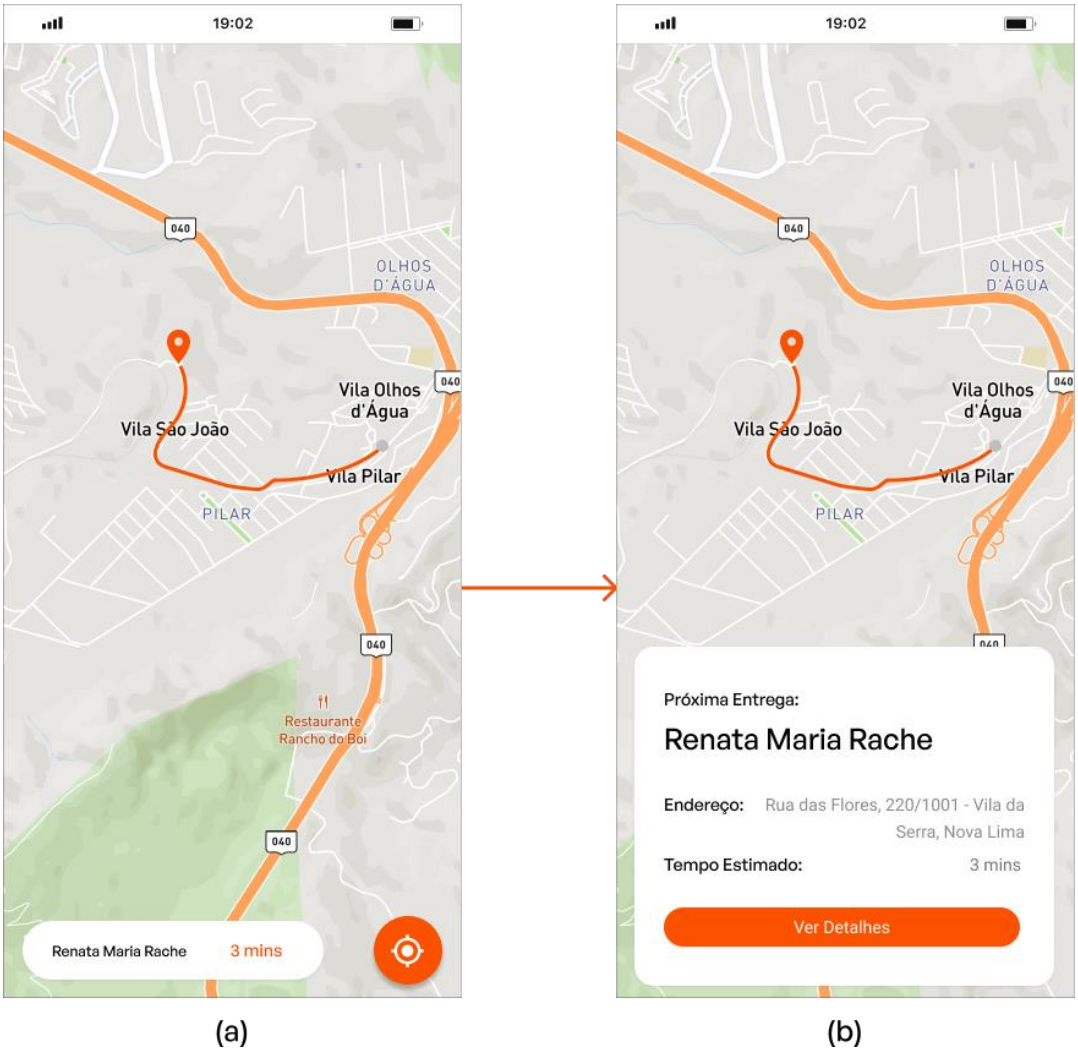


Figura 43. Página de rota de entrega com detalhes minimizados (a) e expandidos (b)

A Figura 44 representa a página de rota de entrega quando a rota está sendo recalculada. Nela é possível ver a posição atual do usuário e uma SnackBar avisando que a rota está sendo recalculada. Esse evento só ocorre no caso do usuário sair da rota pré-definida pelo App, de forma a gerar uma nova rota para ele fazer a entrega.

Figura 44. Página de rota de entrega recarregando rota

A Figura 45 representa a página de detalhes do pedido sendo entregue neste momento. Nela é possível ver: nome e endereço do destinatário e os produtos que devem ser entregues a ele. A partir dessa tela é possível confirmar a entrega ou registrar um problema nela. Desta forma, ela se relaciona com os casos de uso UC9, UC13 e UC14, permitindo visualização dos detalhes do pedido e confirmação da entrega ou registro de problema.

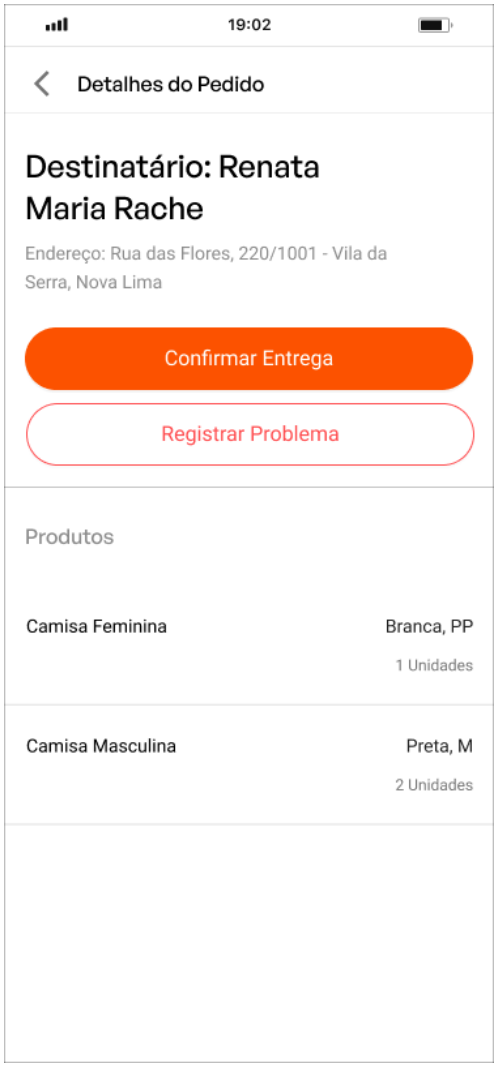


Figura 45. Página de detalhes do pedido

A Figura 46 representa a página de registro de problemas em uma entrega. Ela é um formulário em que o entregador precisa informar qual o tipo de problema, podendo ser destinatário ausente ou produto não disponível, processo representado pela Figura 46 (b). Além do tipo de problema, o entregador pode inserir uma descrição breve do problema. Após preenchimento da tela, ela adquire um estado semelhante ao representado pela Figura 46 (c). Essa página está relaciona aos casos de uso UC14, UC15 e UC16, permitindo o registro de problemas na entrega, relacionados à destinatários ausentes ou falta de produtos.

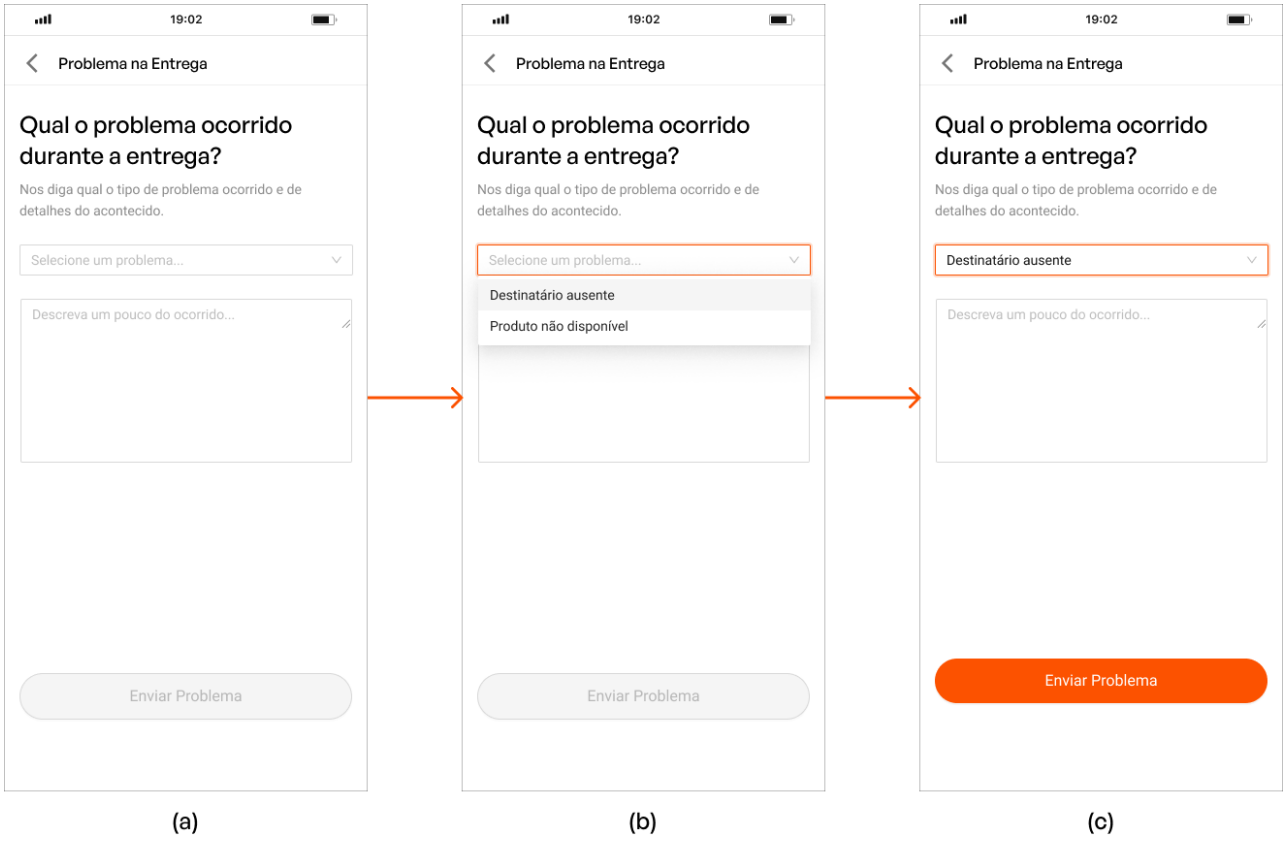


Figura 46. Página de registro de problema na entrega vazia (a), selecionando tipo de problema (b) e preenchida (c)

Após finalizar a entrega de todos os pedidos de uma dada oferta, o entregador é redirecionado à tela representada pela Figura 47. Nela, ele pode ver uma mensagem de agradecimento pela entrega realizada e um botão para que ele seja redireciona à página inicial da aplicação, representada pela Figura 35. Caso o entregador não clique o botão, ele deve ser redirecionado à essa página na próxima vez que abrir a aplicação.

Figura 47. Página de entrega finalizada

4.4 Esboço das Interfaces Usadas pelo Gerente de Operações

A Figura 48 representa o modelo de relatório que deve ser enviado, por Slack ao gerente de operações da empresa, diariamente. Ele deve conter dados relacionados à qual entregador foi responsável, à qual oferta se refere a entrega, e qual produto foi entregue, em qual endereço, assim como o horário específico da entrega e algum problema relacionado a entrega, caso exista. Este relatório se relaciona ao caso de uso UC19.

	A	B	C	D	E	F	G
1	Entregador	Oferta	Produto	Endereço	Hora	Tipo Problema	Obs
2	+5531997865423	6fdf6648-2a1a-4127-aad6-f9d5a7efe32f	312	Rua das Flores, 123, Jardins, Belo Horizonte	01/03/2021 12:30		
3	+5531997865423	6fdf6648-2a1a-4127-aad6-f9d5a7efe32f	413	Rua das Flores, 143, Jardins, Belo Horizonte	01/03/2021 12:35		
4	+5531997865423	6fdf6648-2a1a-4127-aad6-f9d5a7efe32f	541	Rua das Rosas, 23, Jardins, Belo Horizonte	01/03/2021 12:38	Produto em falta	Não foi entregue pelo fornecedor na quantidade certa
5	+5531997865423	6fdf6648-2a1a-4127-aad6-f9d5a7efe32f	325	Rua das Rosas, 56, Jardins, Belo Horizonte	01/03/2021 12:40		
6	+5531997865423	6fdf6648-2a1a-4127-aad6-f9d5a7efe32f	421	Rua das Tulipas, 13, Jardins, Belo Horizonte	01/03/2021 12:45	Remetente ausente	
7	+5531997865423	6fdf6648-2a1a-4127-aad6-f9d5a7efe32f	551	Rua das Tulipas, 53, Jardins, Belo Horizonte	01/03/2021 12:53	Remetente ausente	
8	+5531997563422	011ef997-4001-4aa1-ab1b-99e3879159c1	315	Rua Rio de Janeiro, 153, Centro, Belo Horizonte	01/03/2021 14:05		
9	+5531997563422	011ef997-4001-4aa1-ab1b-99e3879159c1	312	Rua Rio de Janeiro, 53, Centro, Belo Horizonte	01/03/2021 14:15		
10	+5531997563422	011ef997-4001-4aa1-ab1b-99e3879159c1	442	Rua São Paulo, 67, Centro, Belo Horizonte	01/03/2021 14:21		
11	+5531997563422	011ef997-4001-4aa1-ab1b-99e3879159c1	312	Rua São Paulo, 85, Centro, Belo Horizonte	01/03/2021 14:26		

Figura 48. Relatório de entregas realizadas

5. Glossário e Modelos de Dados

Esta Seção tem como objetivo descrever os modelos de dados que compõem a aplicação, assim como definir um glossário que permite interpretar diversos conceitos reservados a este projeto. Tendo isso em vista, a Tabela 4 tem como objetivo definir os diversos atributos que servem de entrada ou saída para aplicação e que são específicos a este projeto. Dessa mesma forma, a Figura 49 tem como objetivo representar a camada de banco de dados da aplicação por meio de um diagrama entidade relacionamento da aplicação. Nesse diagrama é possível identificar todas as tabelas implementadas, assim como a maneira pela qual elas se relacionam.

Autenticação e Acesso à aplicação		
Atributo	**Formato**	**Descrição**
Código de entrega	Número	Código de compartilhamento usado para acessar os dados de uma entrega.
Telefone	Texto	Número de telefone cadastrado no Whatsapp utilizado por ambos entregadores e fornecedores para acessar a aplicação.
Código de 6 dígitos	Número	Código recebido por WhatsApp pelo usuário e que é usado para validar seu telefone durante o processo de autenticação.
Parceiro Zapt	Texto	Sinônimo de fornecedor para o contexto desta aplicação

Processo de Entrega		
Atributo	**Formato**	**Descrição**
Ofertas/Entregas	Objeto	Conjunto de compras com data e rota de entrega a ser realizada.
Endereço inicial	Texto	Endereço pelo qual o entregador deveria iniciar a entrega dos pedidos.
Tempo estimado	Data	Previsão em horas e minutos para que uma entrega seja finalizada seguindo a rota prevista.
Hora de Término	Data	Hora prevista para finalizar a entrega de um pedido específico.
Produtos	Lista	Lista contendo produtos, atributos e suas quantidades necessárias para realizar uma entrega.
Problema na Entrega		
Atributo	**Formato**	**Descrição**
Selecione um problema	Texto	Tipo de problema relacionado à um pedido, podendo ser destinatário ausente ou produto não disponível para entrega
Descreva um pouco o ocorrido	Texto	Descrição extra do problema ocorrido ao realizar a entrega de um pedido.
Relatório de Entregas		
Atributo	**Formato**	**Descrição**
Entregador	Texto	Telefone do entregador responsável pela entrega.
Oferta	UUID	Id relacionado à oferta do pedido entregue.
Produto	Número	Id do produto entregue.
Endereço	Texto	Endereço de entrega do produto.
Hora	Data	Hora exata em que pedido foi assinalado como entregue,
Tipo Problema	Texto	Tipo de problema ocorrido, podendo estar vazio caso nenhum problema tenha ocorrido.
Obs	Texto	Descrição adicional ao problema mencionado caso ele exista.

Tabela 4. Glossário de definições do projeto

Figura 49. Diagrama de Entidade Relacionamento da Aplicação

6. Casos de Teste

Esta seção tem como objetivo descrever os casos de teste previstos para a aplicação. Na Seção 6.1 são apresentados os testes de aceitação referentes às necessidades apresentadas no documento de visão do projeto. Esses testes têm como objetivo garantir que o sistema desenvolvido atende as necessidades básicas de seus usuários. Já na Seção 6.2 é apresentado o plano de teste de integração, que tem como objetivo descrever os testes a serem implementados para garantir que os diversos componentes do sistema funcionem da maneira correta juntos.

6.1 Testes de Aceitação

Nesta Seção 6.1, são descritos os casos de testes relacionados a aceitação do sistema em relação às necessidades que ele busca suprir. Para isso, cada necessidade descrita no documento de visão é descrita por meio de casos de teste representados por meio de uma tabela semelhante à representada pela Tabela 5. Nela são apresentados quatro campos a serem preenchidos. O primeiro é o identificador único referente ao caso de teste proposto. Já o segundo, descreve uma pré-condição que deve ser atendida para que o caso de teste tenha início. Por fim, o terceiro e quarto campos representam as ações tomadas pelos usuários e quais são os resultados esperados como consequência dessas ações. Cada teste descrito foi previamente discutido com o cliente, tendo sido previamente aprovado e estabelecido que a empresa alocará responsáveis por executá-los quando necessário.

Identificador	Teste de aceitação X (TAX)
Pré-condição	Pré condição para caso de teste X
Ações	1. Ação 1 2. Ação 2
Resultados	Resultados

Tabela 5. Exemplo de caso de teste de aceitação

6.1.1 Necessidade 1 – O fornecedor deve ser capaz de visualizar ofertas e atribuir entregadores

A necessidade descrita nesta seção tem como objetivo permitir ao usuário fornecedor visualizar ofertas a serem entregues e compartilhá-las com entregadores para que eles a realizem. Para essa necessidade, foram previstos cinco testes de aceitação, representados pelas Tabela 6, Tabela 7, Tabela 8, Tabela 9 e Tabela 10. Esses casos de teste se relacionam com os casos de uso UC3 e UC4, que preveem que o fornecedor deve ser capaz de ver e compartilhar ofertas a serem entregues e com os casos UC1 e UC2, que preveem que o fornecedor deve ser capaz de se autenticar na aplicação.

Identificador	TA1
Pré-condição	O fornecedor deve possuir um cadastro na aplicação da Zapt
Ações	1. Inserir telefone de acesso 2. Inserir código de validação do telefone
Resultados	O dashboard inicial que contém as listas de ofertas entregues e pendentes de ver mostrado.

Tabela 6. Caso de teste de aceitação 1: Fornecedor deve conseguir acessar a aplicação

Identificador	TA2
Pré-condição	O fornecedor deve estar autenticado na aplicação.
Ações	1. Selecionar a lista de ofertas para hoje
Resultados	O usuário deve ser capaz de visualizar as ofertas que devem ser entregues hoje.

Tabela 7. Caso de teste de aceitação 2: Visualizar ofertas a serem entregues hoje

Identificador	TA3

Pré-condição	O fornecedor deve estar autenticado na aplicação.
Ações	1. Selecionar a lista de ofertas já entregues
Resultados	O usuário deve ser capaz de visualizar as ofertas que já foram entregues

Tabela 8. Caso de teste de aceitação 3: Visualizar histórico de ofertas entregues

Identificador	TA4
Pré-condição	O fornecedor deve estar autenticado na aplicação.
Ações	1. Selecionar a lista de ofertas pendentes
Resultados	O usuário deve ser capaz de visualizar as ofertas que estão pendentes de entrega para outros dias que não hoje.

Tabela 9. Caso de teste de aceitação 4: Visualizar histórico de ofertas pendentes

Identificador	TA5
Pré-condição	O fornecedor deve estar autenticado na aplicação e ter selecionado a lista de ofertas com entrega para hoje.
Ações	3. Selecionar uma oferta da lista 4. Clicar no botão de compartilhamento da oferta
Resultados	Um painel de compartilhamento deve ser aberto para permitir ao fornecedor enviar o código de acesso à oferta ao entregador

Tabela 10. Caso de teste de aceitação 5: Compartilhar código de acesso da entrega com um entregador

6.1.2 Necessidade 2 – A aplicação deve gerar uma rota de entrega otimizada dentro de uma janela de tempo pré-acordada.

A necessidade descrita nesta seção tem como objetivo permitir aos usuários fornecedor e entregador visualizarem uma rota de entrega otimizada das ofertas. Para essa necessidade, foram previstos três testes de aceitação, representados pela Tabela 11, Tabela 12 e Tabela 13. Esses casos de teste se relacionam com os casos de uso UC3, UC10 e UC11, que preveem que uma rota otimizada para realizar as entregas deve ser gerada e armazenada para que ambos fornecedor e entregador possam consultá-la.

Identificador	TA6
Pré-condição	1. Um evento contendo os pedidos feitos em uma oferta deve ser recebido pela oferta e sua rota otimizada deve ser armazenada. 2. O fornecedor deve estar autenticado.
Ações	1. Selecionar a página de ofertas pendentes de entrega 2. Selecionar uma oferta especifica
Resultados	Uma página com a rota de entrega otimizada deve ser disponibilizada.

Tabela 11. Caso de teste de aceitação 6: O fornecedor deve conseguir ver a rota otimizada para uma entrega a ser realizada

Identificador	TA7
Pré-condição	1. Um evento contendo os pedidos feitos em uma oferta deve ser recebido pela oferta e sua rota otimizada deve ser armazenada. 2. O fornecedor deve estar autenticado.
Ações	1. Selecionar a página de ofertas entregues 2. Selecionar uma oferta específica
Resultados	Uma página com a rota de entrega otimizada deve ser disponibilizada.

Tabela 12. Caso de teste de aceitação 7: O fornecedor deve conseguir ver a rota otimizada para uma entrega já realizada

Identificador	TA8
Pré-condição	1. Um evento contendo os pedidos feitos em uma oferta deve ser recebido pela oferta e sua rota otimizada deve ser armazenada. 2. O entregador deve estar autenticado.
Ações	1. Visualizar tela com detalhes da oferta 2. Selecionar mapa com rota otimizada de entrega
Resultados	Uma página com a rota de entrega otimizada deve ser disponibilidade.

Tabela 13. Caso de teste de aceitação 8: O entregador deve conseguir ver a rota otimizada para uma entrega acessada por ele

6.1.3 Necessidade 3 – O entregador deve conseguir acessar e visualizar detalhes da entrega a ser realizada

A necessidade descrita nesta seção tem como objetivo permitir ao entregador acessar os detalhes de uma entrega a ser realizada. Para essa necessidade, foram previstos três testes de aceitação, representados pela Tabela 14, Tabela 15 e Tabela 16. Esses casos de teste se relacionam com os casos de uso UC5, UC6, UC7, UC8, UC9 e UC10, que preveem que o entregador deve ser capaz de acessar uma entrega a partir do seu código de acesso, permitindo a ele visualizar seus detalhes e iniciar o processo de entrega.

Identificador	TA9
Pré-condição	O entregador precisa ter recebido o código de acesso à entrega
Ações	1. Inserir código de entrega 2. Inserir telefone

Resultados	Uma página com a oferta acessada deve ser apresentada.

Tabela 14. Caso de teste de aceitação 9: O entregador deve conseguir acessar a aplicação com um código de entrega

Identificador	TA10
Pré-condição	O entregador deve estar autenticado.
Ações	1. Visualizar tela com detalhes da oferta
Resultados	Uma página com os dados relacionados ao local de início, tempos estimado para realização e produtos a serem entregues da oferta deve ser mostrada.

Tabela 15. Caso de teste de aceitação 10: O entregador deve conseguir ver os detalhes da entrega

Identificador	TA11
Pré-condição	O entregador deve estar autenticado.
Ações	1. Selecionar iniciar entrega 2. Confirmar início 3. Selecionar a opção de ver detalhes do pedido
Resultados	Uma página com a rota de entrega para o pedido atual deve ser mostrada, contendo destinatário, tempo estimado para chegada e pedidos a serem entregues.

Tabela 16. Caso de teste de aceitação 11: O entregador deve conseguir iniciar e seguir a rota de entrega, vendo os detalhes de cada pedido

6.1.4 Necessidade 4 – O entregador deve ser capaz de assinalar entregas e possíveis problemas

A necessidade descrita nesta seção tem como objetivo permitir ao entregador assinalar pedidos como entregues ou relatar possíveis problemas. Para essa necessidade, foram

previstos três testes de aceitação, representados pela Tabela 17, Tabela 18 e Tabela 19. Esses casos de teste se relacionam com os casos de uso UC13, UC14, UC15 e UC16, que preveem que o entregador deve ser capaz de assinalar pedidos como entregues ou registrar problemas, sendo que no segundo caso ele deve ser capaz de registrar problemas relacionados ao destinatário estar ausente ou um produto estar em falta.

Identificador	TA12
Pré-condição	O entregador deve ter iniciado uma entrega
Ações	1. Selecionar a opção de ver detalhes do pedido 2. Selecionar a opção de confirmar entrega
Resultados	Uma mensagem de confirmação de entrega e uma página com a rota de entrega para o próximo destino, devem ser mostradas. Caso seja a última entrega uma página de agradecimento deve ser mostrada.

Tabela 17. Caso de teste de aceitação 12: O entregador deve conseguir assinalar uma entrega como finalizada

Identificador	TA13
Pré-condição	O entregador deve ter iniciado uma entrega
Ações	1. Selecionar a opção de ver detalhes do pedido 2. Selecionar a opção de registrar um problema 3. Preencher formulário de problema selecionando tipo de problema como destinatário ausente.
Resultados	Uma mensagem de confirmação de pedido registrado e uma página com a rota de entrega para o próximo destino, devem ser mostradas. Caso seja a última entrega uma página de agradecimento deve ser mostrada.

Tabela 18. Caso de teste de aceitação 13: O entregador deve conseguir assinalar que um destinatário estava ausente

Identificador	TA14
Pré-condição	O entregador deve ter iniciado uma entrega
Ações	1. Selecionar a opção de ver detalhes do pedido 2. Selecionar a opção de registrar um problema 3. Preencher formulário de problema selecionando tipo de problema como produto faltante.
Resultados	Uma mensagem de confirmação de pedido registrado e uma página com a rota de entrega para o próximo destino, devem ser mostradas. Caso seja a última entrega uma página de agradecimento deve ser mostrada.

Tabela 19. Caso de teste de aceitação 14: O entregador deve conseguir assinalar que um produto estava em falta na hora da entrega

6.1.5 Necessidade 5 – O time de operações deve receber um relatório detalhando as entregas feitas no dia

A necessidade descrita nesta seção tem como objetivo descrever a maneira como o time de operações deve receber um relatório contendo as entregas realizadas nas últimas 24 horas. Para essa necessidade, foram previstos três testes de aceitação, representados pela Tabela 20, Tabela 21 e Tabela 22. Esses casos de teste se relacionam com o caso de uso UC19, que revê que o time de operações deve receber diariamente um relatório detalhado das entregas realizadas.

Identificador	TA15
Pré-condição	Pelo menos uma entrega deve ter sido realizada ao longo do dia
Ações	1. 24 horas se passaram desde o último relatório

Resultados	Um relatório contendo dados detalhados das entregas deve ser enviado à um canal no Slack pré-cadastrado do time de operações.

Tabela 20. Caso de teste de aceitação 15: A cada 24 horas um relatório de entregas deve ser enviado ao time de operações por Slack

Identificador	TA16
Pré-condição	Pelo menos uma entrega deve ter sido realizada ao longo do dia
Ações	1. 24 horas se passaram desde o último relatório
Resultados	Um relatório contendo uma lista de produtos entregues, destinatários, quantidades e horários exatos das entregas realizadas deve ser enviado à um Slack do time de operações.

Tabela 21. Caso de teste de aceitação 16: O relatório deve conter dados detalhados das entregas realizadas

Identificador	TA17
Pré-condição	Pelo menos uma entrega deve ter sido realizada ao longo do dia
Ações	1. 24 horas se passaram desde o último relatório
Resultados	Uma mensagem contendo uma planilha no formato *xlsx* (Formato proprietário da aplicação Excel) deve ser enviada por Slack ao time de operações.

Tabela 22. Caso de teste de aceitação 17: O relatório deve estar em formato compatível com a aplicação Excel

6.2 Testes de Integração

Nesta seção, são descritos os testes de integração propostos para o sistema implementado. Dentro desses testes são representadas interações com aplicações consideradas internas do sistema, essas sendo a Mobile APP e a Delivery, e aplicações externas, como a Shipping

API, Auth API e Ledger API. Para cada caso de teste é usado um modelo de representação tal como o descrito na Tabela 23, esse que representa quais são os sistemas envolvidos, de que maneira eles se comunicam e quais entradas e saídas são esperadas para o teste.

Identificador	Teste de integração Y (TIY)
Sistemas Envolvidos	Sistema 1 e Sistema 2
Interface	Requisição HTTP
Pré-condição	Pré-condição para o caso de teste Y
Entradas	1. Entrada 1 2. Entrada 2
Resultados	Resultados esperados

Tabela 23. Exemplo de caso de teste de integração

Como estratégia para realização dos testes de integração é proposta uma abordagem *top down*. Isso significa dizer que os módulos principais são implementados inicialmente, de forma que para testá-los, *stubs* são implementados para simular o comportamento dos módulos menores ainda não implementados. Para automatização do processo de teste, cada caso de teste na Delivery API será implementado utilizando a biblioteca unittest, essa que é nativa da linguagem python. Já os testes executados pelo Mobile APP serão implementados o pacote integration_test, esse que permitirá a escrita de testes em Dart. Os testes de ambos os componentes serão executados com a utilização do Github Actions, que permite a construção de uma *pipeline* de integração continua no próprio Github. Para sua execução, serão utilizados *mocks* para a base de dados, de forma a evitar a necessidade de implantação de um segundo banco de dados e acelerar a execução dos testes.

Na Tabela 24 é apresentado o caso de teste que representa o fluxo de gerar uma rota de entrega otimizada. Nesse fluxo, a Shipping API se comunica com a Delivery API por meio de eventos enviados por meio de uma fila de mensagens no RabbitMQ. Esse evento contém

dados relacionados aos pedidos, suas ofertas e em quais endereços eles devem ser entregues. Ao receber esse evento, a aplicação gera uma rota otimizada para realizar as entregas e a armazena para futuras consultas.

Identificador	TI1
Sistemas Envolvidos	Shipping API e Delivery API
Interface	Mensagem RabbitMQ
Pré-condição	Uma oferta precisa ter tido sucesso ao atingir o mínimo de compras necessário.
Entradas	1. Lista de pedidos feitos para uma da oferta
Resultados	Deve ser extraído o endereço de cada pedido de forma a gerar uma rota otimizada para que as entregas sejam realizadas.

Tabela 24. Caso de teste de integração 1: Testar gerar rota de entrega otimizada

Na Tabela 25 é representado um caso de teste que tem como objetivo testar o processo de autenticação do fornecedor na aplicação. Nesse, o usuário interage por meio da Mobile APP, inserindo o telefone e o código de validação enviado por WhatsApp. Para seu funcionamento o Mobile APP se comunica com a Delivery API por meio de requisições HTTP. A Delivery API por sua vez deve se comunicar com a Auth API a fim de autenticar o fornecedor e recuperar seus dados. Como resultado esperado do caso de teste, o fornecedor deve se encontrar autenticado e na página inicial da aplicação, essa que contém as listas das ofertas a serem entregues.

Identificador	TI2
Sistemas Envolvidos	Mobile APP, Delivery API e Auth API
Interface	Requisições HTTP

Pré-condição	O telefone utilizado no teste deve estar cadastrado como fornecedor na aplicação.
Entradas	1. Telefone utilizado para autenticação do fornecedor 2. Código de validação do telefone enviado por WhatsApp
Resultados	Caso o telefone esteja cadastrado e o código esteja correto o fornecedor deve se encontrar autenticado e na página inicial da aplicação.

Tabela 25. Caso de teste de integração 2: Testar autenticação do fornecedor na aplicação

Na Tabela 26 é representado um caso de teste que tem como objetivo testar o processo de compartilhamento de uma entrega por um fornecedor. Nesse, o usuário interage por meio da Mobile APP, selecionando uma oferta e selecionando a opção de compartilhá-la com um entregador. A aplicação deve ser capaz de retornar os dados da oferta assim como seu código de acesso, para que o fornecedor possa compartilhá-la.

Identificador	TI3
Sistemas Envolvidos	Mobile APP e Delivery API
Interface	Requisições HTTP
Pré-condição	O fornecedor deve estar autenticado na aplicação.
Entradas	1. Seleção de uma entrega da lista 2. Clique no botão de compartilhar entrega
Resultados	Um *ShareIntent* deve ser apresentado ao usuário solicitando que ele selecione com que aplicativo compartilhar o código de acesso da oferta selecionado.

Tabela 26. Caso de teste de integração 3: Testar compartilhamento de entregas pelo fornecedor

Na Tabela 27 é representado um caso de teste que tem como objetivo testar o processo de autenticação de um entregador em uma oferta. Nesse, o usuário interage por meio da Mobile APP, inserindo um código de acesso recebido e seu telefone. A aplicação deve ser capaz de verificar se uma oferta existe para o telefone informado, e caso exista, solicitar ao entregador que ele insira seu telefone para que ele seja armazenado.

Identificador	TI4
Sistemas Envolvidos	Mobile APP e Delivery API
Interface	Requisições HTTP
Pré-condição	O entregador deve possuir um código de acesso para uma oferta
Entradas	1. Código de acesso de uma oferta 2. Telefone de acesso do entregador
Resultados	O entregador deve ser autenticado e redirecionado para a tela contendo os detalhes da oferta acessada

Tabela 27. Caso de teste de integração 4: Testar autenticar entregador

Na Tabela 28 é representado um caso de teste que tem como objetivo testar o processo entrega de pedidos de uma entrega. Nesse, o usuário interage por meio da Mobile APP, iterando por cada pedido ao marcá-los como entregues ou como com problema. A aplicação deve mostrar cada pedido à medida que o anterior é finalizado, mostrando também o caminho de entrega para o próximo destinatário. Por fim, uma mensagem deve ser enviada a fim de notificar a Ledger API de que todos os pedidos foram entregues.

Identificador	TI5
Sistemas Envolvidos	Mobile APP, Delivery API e Ledger API
Interface	Requisições HTTP e Mensagem RabbitMQ

Pré-condição	O entregador deve estar autenticado na aplicação
Entradas	1. Entrega deve ter sido iniciada 2. Selecionar a opção de confirmar entrega ou registrar problema para cada pedido da entrega.
Resultados	O entregador deve ser direcionado para os endereços de cada pedido à medida que ele marca um pedido como entregue ou com problema. Ao finalizar a entrega, uma mensagem deve ser enviada por RabbitMQ informando da sua finalização.

Tabela 28. Caso de teste de integração 5: Testar entrega dos pedidos

Na Tabela 29 é representado um caso de teste que tem como objetivo testar o processo de envio diário de um relatório de entregas ao time de operações. Nesse, a Delivery API deve ser acionada pelo seu *job* responsável por solicitar o envio do relatório a cada 24 horas. A aplicação deve ser capaz de recolher dados relacionados às entregas realizadas e escrevê-los em um arquivo Xlsx representado pela Figura 48. Após escrita do arquivo, esse deve ser enviado ao Slack do time de operações.

Identificador	TI6
Sistemas Envolvidos	Delivery API
Interface	Requisições HTTP
Pré-condição	Entregas devem ter sido realizadas nas últimas 24 horas
Entradas	1. O *job* responsável por solicitar o envio do relatório deve ser acionado devido ao tempo
Resultados	Uma mensagem contendo o relatório de entregas deve ser enviado à um canal de Slack.

Tabela 29. Caso de teste de integração 6: Testar relatório de entregas diário

7. Cronograma e Processo de Implementação

Nesta seção, são descritos o cronograma de desenvolvimento da aplicação e seu processo de implementação. A Seção 7.1 contém o cronograma de implementação do sistema proposto, esse contém as tarefas a serem implementadas ao longo dos 4 meses e meio de desenvolvimento do projeto. Já a Seção 7.2 contém os detalhes de como essa implementação será realizada, destacando práticas aplicadas e artefatos produzidos.

7.1 Cronograma

O período para desenvolvimento do projeto proposto é de cerca de quatro meses e meio, tendo seu início no dia 1 de fevereiro de 2022 e seu fim no dia 15 de maio de 2022. Tendo isso em vista o cronograma é dividido em períodos, denominados de Milestones, tendo como objetivo permitir um *feedback* constante do cliente para com o produto. Na Tabela 30 são detalhadas cada Milestone e qual seu período previsto. Dessa mesma forma, também são detalhadas quais atividades são propostas para cada período identificado.

Período	Atividades

Milestone 1
01/02 – 27/02
	• Desenvolvimento da estrutura inicial da aplicação móvel:
	  ○ Estrutura de pastas
	  ○ Tema da aplicação
	  ○ Instalação de bibliotecas
	  ○ Preparação das variáveis de ambiente
	• Desenvolvimento da estrutura inicial da API:
	  ○ Estrutura de pastas
	  ○ Instalação de bibliotecas
	  ○ Imagem Docker
	  ○ Preparação das variáveis de ambiente
	• Implementação das estruturas de integração e implantação contínuas.
	• Disponibilização das aplicações por meio da Heroku/ Codemagic:
	  ○ Configurações necessárias para execução da API na Heroku

	○ Disponibilização da aplicação móvel por meio da Google Play Store e Apple App Store • Implementação das tabelas do banco de dados e seus modelos de dados. • Implementação do fluxo[1] de gerar rota de entrega otimizada representado pela Figura 17: ○ Realização dos casos de teste de aceitação TA6, TA7 e TA8. ○ Implementação do caso de teste de integração TI1. • Implementação dos componentes e telas relacionados ao fluxo de autenticação de fornecedores e entregadores. • Casos de uso relacionados: UC1, UC2, UC3, UC5, UC6, UC11, UC12, UC10 e UC20.
Milestone 2 **28/02 – 20/03**	• Correções com relação ao *feedback* da Sprint passada. • Implementação do fluxo de autenticação do fornecedor e entregador representados pela Figura 18 e Figura 20: ○ Realização dos casos de teste de aceitação TA1 e TA9. ○ Implementação dos casos de teste de integração TI2 e TI4. • Casos de uso relacionados: UC1, UC2, UC5 e UC6. • Implementação do fluxo de compartilhamento de entregas representado pela Figura 19: ○ Realização dos casos de teste de aceitação TA2, TA3, TA4, TA5. ○ Implementação do caso de teste de integração TI3. • Casos de uso relacionados: UC4.

[1] Por implementação de um fluxo entende-se a realização das seguintes atividades:
- Implementação das telas e componentes relacionados ao fluxo
- Implementação das classes e métodos necessários para funcionamento do fluxo

Milestone 3 **21/03 – 10/04**	• Correções com relação ao *feedback* da Sprint passada. • Início da implementação do fluxo de entrega de pedidos representado pela Figura 21: ○ Nessa Sprint não é necessário a implementação dos Mapas e roteamento de entregas. • Casos de uso relacionados: UC7, UC8, UC9, UC10, UC13, UC14, UC15, UC16 e UC17.

Milestone 4
11/04 – 01/05
	• Correções com relação ao *feedback* da Sprint passada.
	• Finalização da implementação do fluxo de entrega de pedidos representado pela Figura 21:
	  ○ Implementação dos Mapas e roteamento de entrega.
	  ○ Realização dos casos de teste de aceitação TA10, TA11, TA12, TA13 e TA14.
	  ○ Implementação do caso de teste de integração TI5.
	• Casos de uso relacionados: UC7, UC8, UC9, UC10, UC13, UC14, UC15, UC16 e UC17.

Milestone 5
02/05 – 15/05
	• Correções com relação ao *feedback* da Sprint passada
	• Implementação do fluxo de envio de um relatório de entregas diário representado pela Figura 22:
	  ○ Realização dos casos de teste de aceitação TA15, TA16 e TA17.
	  ○ Implementação do caso de teste de integração TI6
	• Ajustes finais e *deploy* final da aplicação para as lojas de aplicativo e da API para AWS.
	• Casos de uso relacionados: UC19
	• Post-mortem do projeto

Tabela 30. Cronograma de atividades do projeto

7.2 Processo de Implementação

O objetivo desta seção é descrever o processo de implementação a ser seguido durante o desenvolvimento do sistema proposto. Tendo isso em vista, o primeiro ponto a ser mencionado é que esse projeto tem como objetivo seguir um método de desenvolvimento incremental com Sprints de desenvolvimento com aproximadamente 15 dias de duração. Durante cada ciclo foram determinadas atividades a ser realizadas conforme descrito na Seção 7.1.

Para cada ciclo de desenvolvimento é proposto a implementação de um dos fluxos dos usuários na aplicação. Contudo, a primeira Sprint é composta apenas pela configuração dos sistemas e suas estruturas iniciais, não prevendo a implementação de nenhum fluxo, apenas das tabelas do banco de dados. Nesse primeiro momento, também são feitas as configurações iniciais relacionadas à implantação da API na AWS e à disponibilização do aplicativo móvel nas lojas de aplicativos Google Play Store e Apple App Store. Após a fase inicial do projeto, se seguem mais 6 sprints que contém os seguintes aspectos:

- Correção e ajustes realizados devido à possíveis comentários feitos com relação as entregas realizadas na Sprint anterior.
- Implementação de um fluxo da aplicação, esse contendo as páginas necessárias no Mobile APP e as classes e métodos na Delivery API, assim como o estabelecimento da comunicação entre os dois.
- Realização dos testes de aceitação relacionados aos fluxos desenvolvidos na Sprint.
- Implementação dos testes de integração relacionados aos fluxos desenvolvidos na Sprint.

Devido ao tamanho do processo de entrega de pedidos, representado pelo diagrama de sequência da Figura 21, esse fluxo foi quebrado em duas Sprints, sendo que apenas no segundo são implementados seus testes.

Após a última iteração do projeto, representada pela Sprint 7, este se dá por terminado. Com seu fim, se faz necessário a entrega desta Documentação em conjunto com o Documento de Visão e os diagramas desenvolvidos. Dessa mesma forma, o código de todo o sistema deve ser disponibilizado em um repositório aberto no Github.

8. Post-mortem

Nesta seção, é descrito um relato de experiências obtidas ao desenvolver este projeto. A Seção 8.1 contém o um resumo das experiências positivas percebidas. A Seção 8.2 apresenta algumas experiências negativas. Por fim, a Seção 8.3 contém algumas lições aprendidas durante a implementação da aplicação proposta.

8.1 Experiência Positivas

Das experiências positivas, algumas recebem uma ênfase maior. Essas referem-se a escolha das tecnologias Flutter, para desenvolvimento do aplicativo *mobile*, e Python, para implementação da API, que proporcionou um desenvolvimento simples e eficiente, utilizando padrões modernos e que poderão ser reutilizados pelo cliente futuramente. Simultaneamente, a utilização do Github Actions para construção do *pipeline* de CI/CD da aplicação foi extremamente positiva, permitindo que os testes fossem executados de maneira recorrente, além de garantir um *deploy* rápido para o ambiente de desenvolvimento. Por fim, o planejamento completo da aplicação previamente ao desenvolvimento permitiu que a fase de implementação fosse melhor planejada, apresentando poucas surpresas e se tornando mais previsível.

8.2 Experiência Negativas

Durante a implementação da solução proposta não foram encontradas tantas experiências negativas. Entretanto, duas podem ser citadas como tendo atrapalhado de alguma forma o projeto. A primeira foi a escolha inicial da plataforma Codemagic para implementação da *pipeline* de CI/CD do aplicativo mobile. Essa plataforma, por mais que apresente de fato diversas vantagens, ainda apresenta a falta de algumas funcionalidades consideradas necessárias para o projeto, como a possibilidade de execução de casos de teste apenas quando determinadas pastas são modificadas. Simultaneamente, o emulador do sistema operacional Android para Windows apresentou alguns problemas frequentes quanto à funcionalidade de geolocalização. Por diversas vezes ele não registrava a localização informada ou simplesmente não permitia que essa funcionalidade fosse utilizada, necessitando que o emulador fosse recriado algumas vezes.

8.3 Lições Aprendidas

Da perspectiva de lições aprendidas, é interessante observar que mesmo tendo planejado o projeto em detalhes e com antecedência, ainda se faz necessário modificar diversas partes desse. Isso se fez necessário devido às regras de negócio que foram modificadas, aprendizado do desenvolvedor envolvido ou detalhes que possam ter passado despercebido durante o planejamento inicial e levantamento de requisitos. Dessa forma, fica ainda mais claro a necessidade de uma metodologia de desenvolvimento ágil para a implementação de sistemas para *startups*, como o cliente deste projeto. Derivado dessa necessidade, também surgiu a necessidade de manter este documento sempre atualizado, de forma que tarefas precisaram ser reservadas com este foco, algo que geralmente pode não ser feito em um contexto diferente, como o de mercado de trabalho.

8.4 Repositório do Trabalho

O código fonte deste projeto, assim como os demais artefatos produzidos durante o desenvolvimento deste trabalho, pode ser encontrado no repositório do Github a seguir: https://github.com/ICEI-PUC-Minas-PPLES-TI/plf-es-2021-2-tcci-5308100-dev-aylton-almeida.