



COMMODITY
SUPPLIES

CSA tabex system

Field Rules Control

FIELD RULES CONTROL - TABEX

CSA Tabex System



VERSION 1.05

JULY/2020

COMMODITY SUPPLIES AG – CSA DO BRASIL

FIELD RULES CONTROL - TABEX

© COMMODITY SUPPLIES AG – CSA DO BRASIL
Alameda dos Nhambiquaras, 1770 cj 108, Moema
São Paulo - SP Brasil - CEP: 04090-004
Telephone: +55 11 3040-4782 • www.commodity.ch



DOCUMENT VERSION CONTROL

Version	Date	Alterations	Person in Charge
1.00	03/18/2020	Document Creation	Luciana Luvisotto
1.01	03/20/2020	Creation of the Booking Plan Module	Bruna Budri
1.02	03/20/2020	Creation of the Logistics Plan Module	Bruna Budri
1.03	03/27/2020	Creation of the VGM + Draft Module	Bruna Budri
1.04	04/02/2020	Inclusion of images	Bruna Budri
1.05	07/24/2020	Manual Update	Luciana Luvisotto

GENERAL GUIDELINES TO ACCESS THE TABEX SYSTEM

The objective of the current document is to guide the use of the Tabex System and available services.

1 - There are 4 Access levels to the system, namely:

- 1.1 Level 1 for users with full access to the system except the development area. Access will be granted for consultation, editing, registration and exclusion.
- 1.2 Level 2 for users with access granted for consultation, editing and registration.
- 1.3 Level 3 for users with access granted for consultation and registration.
- 1.4 Level 4 for users with access granted for consultation only.

2 - The development department must provide the master user (level 1 access) to register the other necessary users.

Summary

COPYRIGHTS	6
CONTACT INFORMATION CSA DO BRASIL.....	6
CSA AROUND THE WORLD.....	6
1. LOADING PLAN.....	7
1.1 LOADING LIST (OPERATIONAL DASHBOARD).....	8
1.2 PENDING LIST.....	8
1.3 CANCELLED LIST	9
1.4 BOOKING PLAN.....	9
1.5 LOGISTICS PLAN	11
1.6 SHIPPED LIST.....	14
1.7 AÇÕES GERAIS PARA BOOKING PLAN E LOGISTICS PLAN	15
1.8 DICIONÁRIO DE DADOS SIMPLIFICADO E TABLES ESSENCIAIS.....	22
2. USERS	26
3. MENU	30
4. USERS X MENU	33
5. VESSEL	34
6. CARGO TYPES	36
7. CARRIER.....	39
8. CONTAINER TYPE.....	42
9. LOADING PORTS.....	45
10. PRODUCTS	48
11. TANK OWNER.....	52
12. ANEXOS TYPES	55
13. DESTINATION PORTS	58
14. COUNTRIES	61
15. U.N.	63
16. ROUTES.....	66
17. CLIENTS.....	70
18. BANKING DETAILS	76
19. ALLOCATION	79
20. FLEXI STOCK	82
21. FINANCE.....	86
22. PAYMENT REQUEST - ROUTINE	89
23. TARIFF COSTS	94
24. WEBSERVICE ROUTINE.....	94
 GLOSSARY	95

Copyrights

Copyright © 2020 COMMODITY SUPPLIES AG – CSA do Brasil, with headquarters located at Alameda dos Nhambiquaras, 1770 CJ. 108, Moema, São Paulo - SP Brazil - Zip Code: 04090-004. All rights reserved. This manual or any part of it may not be reproduced or otherwise used without the express written permission of the author or publisher, except for the use of brief quotations in a review of the manual. All rights reserved. First Edition. July | 2020.
www.commodity.ch

Contact Information CSA do Brasil

COMMODITY SUPPLIES AG – CSA do Brasil
Alameda dos Nhambiquaras, 1770 cj
108, Moema. São Paulo - SP Brasil - CEP:
04090-004

CSA around the World

COMMODITY SUPPLIES AG
Roosstrasse 53, CH-8832 Wollerau
Suíça. +41 44 807 44 44 info@commodity.ch
Brazil | Germany | United States of America
Singapore | Costa Rica | China

1. Loading Plan

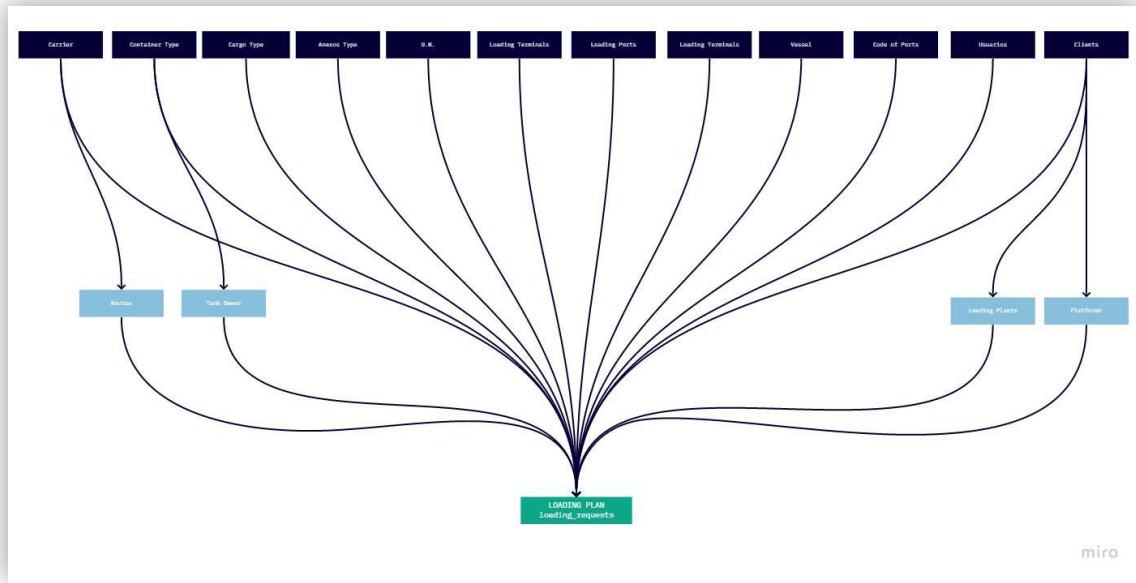


Figure 1 – General View

Description: Contemplates the beginning of the entire logistics programming of CSA. From the moment a programming is included, the user will have, intuitively, all the necessary steps to be followed during the whole process.

Booking Plan Flow

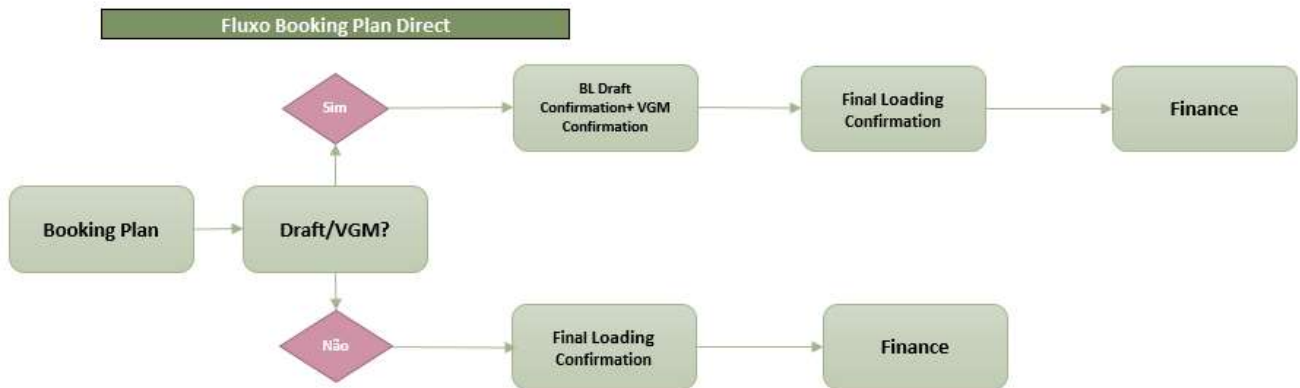


Figure 2 – Booking Plan Flow

Logistics Plan Flow

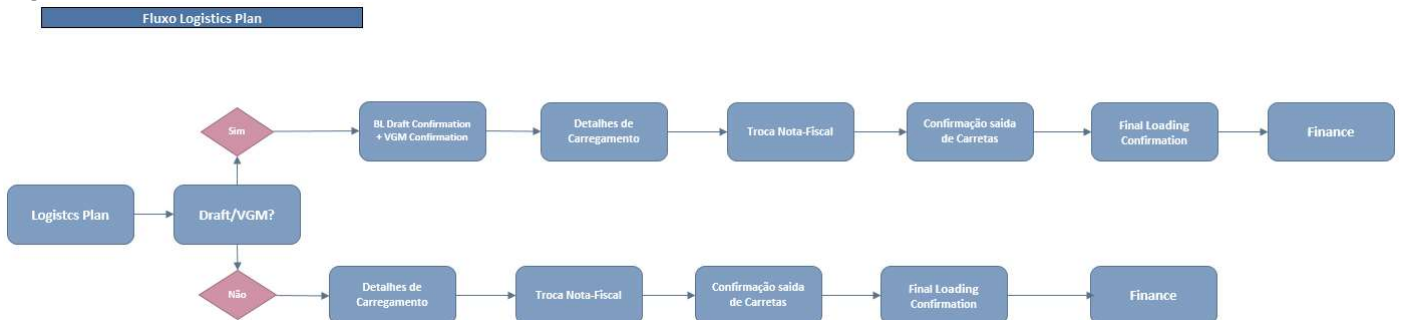


Figure 3 - Logistics Plan Flow

Pre-Conditions: Secondary registrations:

- | | | |
|--------------------------------|-------------------------|------------------|
| ✓ Shippers (Client) | ✓ Hauling (Client) | ✓ Code of Ports |
| ✓ Loading Terminal (Client) | ✓ Client | ✓ Container Type |
| ✓ Tank Owner (Client) | ✓ Carriers | ✓ Cargo Type |
| ✓ Empty Release Place (Client) | ✓ Product | ✓ U.N. |
| ✓ Stuffing Place (Client) | ✓ Vessel | ✓ Anexos Type |
| ✓ Freight/Forwarder (Client) | ✓ Loading Port (Brazil) | ✓ Routes |

Subsidy to perform the following:

- ✓ VGM
- ✓ Draft BL
- ✓ Loading Confirmation (departed)
- ✓ Allocation
- ✓ Finance

1.1 Loading List (Operational Dashboard)

Shows all Booking Plan and Logistics Plan with 'On Hold' Status in list format.

In this screen the user will have the following options: Consult individually for greater detail Edit, Cancel, Clone, Link to other canceled plans, Send logistics schedule, report Draft BL status, VGM status and send shipment confirmation (BL).

Through this screen the user can search by the client trade name, name of the user responsible for the registration of the logistic plan and Booking number.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/list
HTTP Method:	GET
Sending Parameters:	?tipo=loading
Internal Information:	
View:	'loading.loadingPlan.list'
View Parameters:	compact('registers')
Method:	LoadingController@index
Mandatory Folder:	loading/loadingPlan/*
Route Name:	listLoading

1.2 Pending List

Displays all Booking Plan and Logistics Plan with Status 'Pending' in list format.

Through this screen the user will have the option to edit and remove the record from the pending list as soon as the record is updated.

Through this screen it is possible to search by client fantasy name, name of user responsible for the registration of the logistic plan and booking number.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/pendingList
HTTP Method:	GET
Sending Parameters:	?tipo=pending
Internal Information:	
View:	'loading.loadingPlan.list'
View Parameters:	compact('registers')
Method:	LoadingController@index
Mandatory Folder:	loading/loadingPlan/*
Route Name:	PendingList

1.3 Cancelled List

Presents all Booking Plan and Logistics Plan with Status 'Cancel' in list format. This list is presented for consultation only, and the user cannot perform any actions on the records.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/cancelled
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	loading.loadingPlan.cancelledlist
View Parameters:	compact('registers')
Method:	LoadingController@cancelled
Mandatory Folder:	loading/loadingPlan/*
Route Name:	PendingList

1.4 Booking Plan

Responsible for the entire logistics chain of Booking Plan, sending the BL Draft, VGM and Loading Confirmation.

Available actions to load this chain are:

1.4.1 Create

Form to insert a new Booking Plan record.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/bk/create
HTTP Method:	GET
Sending Parameters:	?tipo=loadingBk
Internal Information:	
View:	loading.loadingPlan.createBk
View Parameters:	compact('select','attachments')
Method:	LoadingController@createBk
Mandatory Folder:	loading/loadingPlan/*
Route Name:	createBookingPlan

1.4.1.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	LoadingController@store
Route Name:	storeLoading

1.4.2 Edit

Action of the system to edit the Booking Plan record.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/ bk/{id}/edit
HTTP Method:	GET
Sending Parameters:	id

Internal Information:	
View:	loading.loadingPlan.editBk
View Parameters:	compact('register','select','attachments','usuario')
Method:	LoadingController@edit
Mandatory Folder:	loading/loadingPlan/*
Route Name:	editBookingPlan

1.4.2.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/{id}/update
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	LoadingController@update
Route Name:	updateLoading

1.4.3 Clone

This action clones the desired record to facilitate the work and optimize the user's time.

Through this action, the only field not cloned is the Booking Number, which as a rule, must necessarily be different from the numbers previously registered.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/bk/{id}/clone
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	loading.loadingPlan.editBk
View Parameters:	compact('register','select','attachments','usuario')
Method:	LoadingController@clone
Mandatory Folder:	loading/loadingPlan/*
Route Name:	cloneBookingPlan

1.4.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/{id}/update
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	LoadingController@update
Route Name:	updateLoading

1.4.4 Send an E-mail

This action sends the e-mail with the Logistic Programming.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/emailView/{id}
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	mail_templates.previouslyEmail

View Parameters:	compact('loading', 'attachments', 'bl')
Method:	MailSendController@previouslyEmail
Mandatory Folder:	mail_templates/*
Route Name:	previouslyEmail
Controller:	MailSendController
Controller (Mail):	App\Mail\BookingConfirmation

1.4.4.1 Email Pre-view

Item	Value
Internal Information:	
View:	mail_templates.mailBookingConfirmation
View Parameters:	compact('loading', 'attachments', 'select', 'dePara', 'loading_empties', 'item')
Method:	MailSendController@previouslyEmail
Mandatory Folder:	mail_templates/*

1.4.4.2 E-mail Send Controller

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/sendemail/{id}
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
Method:	MailSendController@sendEmail
Mandatory Folder:	mail_templates/*
Route Name:	sendEmail
Controller (Mail):	App\Mail\BookingConfirmation

1.5 Logistics Plan

Responsible for the entire logistics chain of Booking Plan, sending the BL Draft, Iso/FlexiTank Plan, Equipment Removal, Stuffing Terminal, VGM and Loading Confirmation.

Available actions to load this chain are:

1.5.1 Create

Form to insert a new Loading Logistics record.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/loadingPlan/create
HTTP Method:	GET
Sending Parameters:	tipo=loading
Internal Information:	
View:	loading.loadingPlan.create
View Parameters:	compact('select', 'attachments')
Method:	LoadingController@create
Mandatory Folder:	loading/loadingPlan/*
Route Name:	createLoading
Controller:	LoadingController

1.5.1.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	LoadingController@store
Route Name:	storeLoading
Controller:	LoadingController

1.5.2 Edit

Action of the system to edit the Booking Plan record.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/loadingPlan/{id}/edit
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	loading.loadingPlan.edit
View Parameters:	compact('register','select','attachments','usuario','loading_stuffings','loading_empties')
Method:	LoadingController@edit
Mandatory Folder:	loading/loadingPlan/*
Route Name:	editLoading
Controller:	LoadingController

1.5.2.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/{id}/update
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	LoadingController@update
Route Name:	updateLoading
Controller:	LoadingController

1.5.3 Clone

This action clones the desired record to facilitate the work and optimize the user's time.

Through this action, the only field not cloned is the Booking Number, which as a rule, must necessarily be different from the numbers previously registered.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/loadingPlan/{id}/clone
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	loading.loadingPlan.edit
View Parameters:	compact('register','select','attachments','usuario','loading_stuffings','loading_empties')

Method:	LoadingController@clone
Mandatory Folder:	loading/loadingPlan/*
Route Name:	cloneLoading
Controller:	LoadingController

1.5.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/{id}/update
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	LoadingController@update
Route Name:	updateLoading
Controller:	LoadingController

1.5.4 Send an E-mail

This action sends the e-mail with the Logistics Programming.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/emailView/{id}
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	mail_templates.previouslyEmail
View Parameters:	compact('loading', 'attachments','bl')
Method:	MailSendController@previouslyEmail
Mandatory Folder:	mail_templates/*
Route Name:	previouslyEmail
Controller:	MailSendController
Controller (Mail):	App\Mail\BookingConfirmation

1.5.4.1 Email Pre-view

Item	Value
Internal Information:	
View:	mail_templates.mailLogisticsConfirmation
View Parameters:	compact('loading', 'attachments','loading_stuffings','loading_empties', 'select','dePara')
Method:	MailSendController@previouslyEmail
Mandatory Folder:	mail_templates/*







1.5.4.2 E-mail Send Controller

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/sendemail/{id}
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
Method:	MailSendController@sendEmail
Mandatory Folder:	mail_templates/*
Route Name:	sendEmail
Controller (Mail):	App\Mail\BookingConfirmation

1.6 Shipped List

This list displays all Booking Plans and Logistics Plans with “Shipped” Status in list format.

Through this screen the user will have the option to view the record, and will be able to:

- Resend the Loading Confirmation e-mail 
 - Clone a Record for a New Logistics Process 
 - Send Request for Payment of Fees and Local Services Provided to the Client 
 - Download the records displayed on screen 
 - View the BL information through the Icon 
 - View the process with the main process information using the icon: 
- Hovering the mouse over the mentioned icon will display a pop-up similar to the image below.

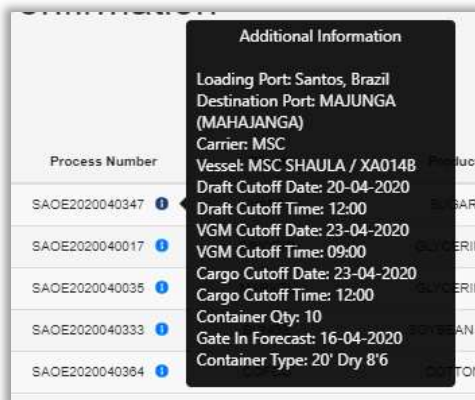


Figure 4 – Detailed process information


Searches can be performed by:

- Date period (BL date).
- Fantasy Name of Client
- User Name
- SAOE
- Booking Number
- Vessel

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingPlan/list
HTTP Method:	GET
Sending Parameters:	?tipo=confirmation
Internal Information:	
View:	'loading.loadingPlan.list'
View Parameters:	compact('registers','tipo','totais')
Method:	LoadingController@index
Mandatory Folder:	loading/loadingPlan/*
Route Name:	listLoading

1.7 General Actions for Booking Plan and Logistics Plan

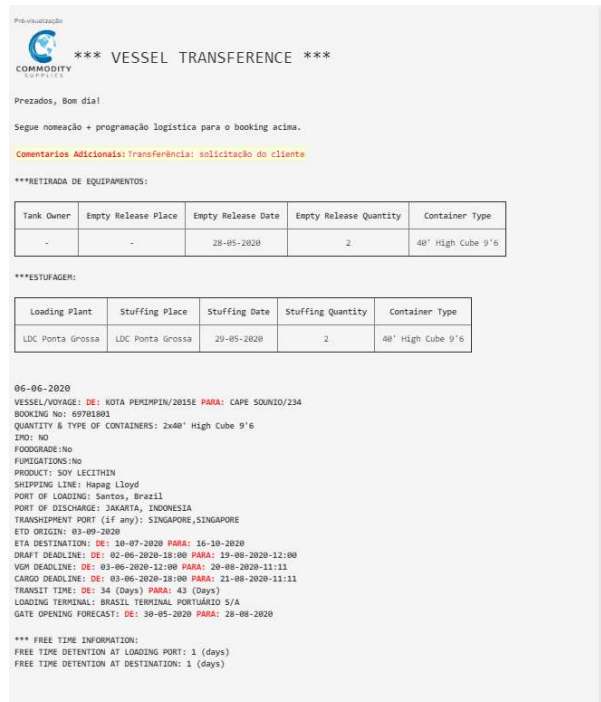
1.7.1 Transfer To Another Booking

By clicking on the icon  available on the Operational Dashboard / Allocation. The user must insert the reasons to transfer the Booking.

This action creates a Booking log with previous data, creating a record in "Loading_requests" with Status 'Transfer'. After the logo, another record with Status 'Pending' is generated with null data from the logistic programming, sending only the number for Booking, Client, Shipper, Product, Carrier, Route and Stuffing Plan.

The record is available in the 'Pending List' menu until the user responsible for the record adds the new programming information.

As soon as the user adds the new information, the user is redirected to the e-mail screen where all the old and new information of the respective Booking will be displayed.



06-06-2020
***** VESSEL TRANSFERENCE *****
 Prezados, Bom dia!
 Segue nomeação + programação logística para o booking acima.
 Comentários Adicionais: Transfêrencia: solicitação do cliente
 ***RETIRADA DE EQUIPAMENTOS:

Tank Owner	Empty Release Place	Empty Release Date	Empty Release Quantity	Container Type
-	-	28-05-2020	2	40' High Cube 9'6

 ***ESTUFAGEM:

Loading Plant	Stuffing Place	Stuffing Date	Stuffing Quantity	Container Type
LDC Ponta Grossa	LDC Ponta Grossa	20-05-2020	2	40' High Cube 9'6

 06-06-2020
 VESSEL/VOYAGE: DE: KOTA PEMIMPIN/2015E PARA: CAPE SOUND/234
 BOOKING No: 69781801
 QUANTITY & TYPE OF CONTAINERS: 2x40' High Cube 9'6
 JNO: NO
 FOODGRADE:No
 FUMIGATIONS:No
 PRODUCT: SOY LECITHIN
 SHIPPING LINE: Mapag Lloyd
 PORT OF LOADING: Santos, Brazil
 PORT OF DISCHARGE: JAKARTA, INDONESIA
 TRANSHIPMENT PORT (if any): SINGAPORE, SINGAPORE
 ETD ORIGIN: 03-09-2020
 ETA DESTINATION: DE: 10-07-2020 PARA: 16-10-2020
 DRAFT DEADLINE: DE: 02-06-2020-18:00 PARA: 19-08-2020-12:00
 VGR DEADLINE: DE: 03-06-2020-12:00 PARA: 20-08-2020-11:11
 CARGO DEADLINE: DE: 03-06-2020-18:00 PARA: 21-08-2020-11:11
 TRANSIT TIME: DE: 34 (Days) PARA: 43 (Days)
 LOADING TERMINAL: BRASIL TERMINAL PORTUARI20 5/A
 GATE OPENING FORECAST: DE: 30-05-2020 PARA: 28-08-2020
 *** FREE TIME INFORMATION:
 FREE TIME DETENTION AT LOADING PORT: 1 (days)
 FREE TIME DETENTION AT DESTINATION: 1 (days)

Figure 5 – Screen for Vessel Transference

1.7.2 Link to Another Booking


This action is responsible for creating a link between Booking Plans.

By using this action, the user will keep a history which will later be displayed in Reports. 

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/link/create
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/modal_link.blade.php
View Parameters:	data-catid = catid
Method:	LinkController@create
Route Name:	linkBooking
Controller:	LinkController

1.7.3 Update Draft BL + VGM

Draft BL

This action is responsible for updating the Draft BL sending process. 

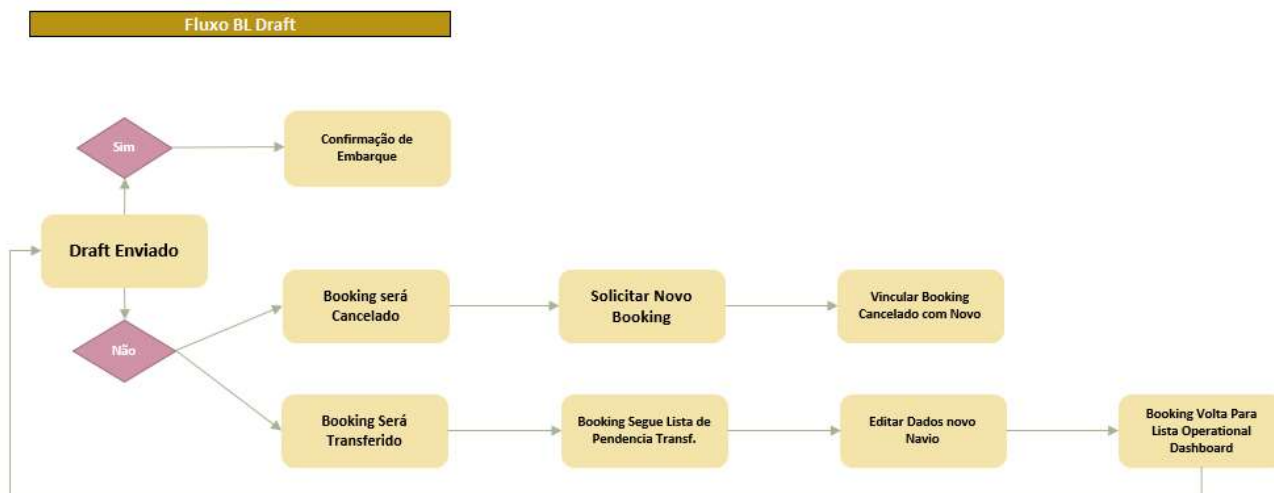


Figure 6 –BL Draft Flow

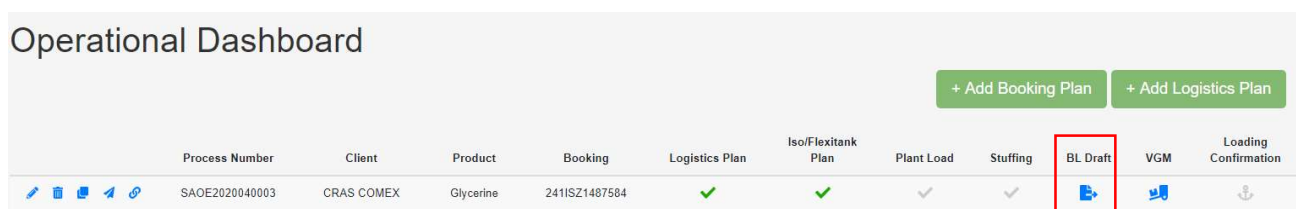


Figure 7 – “BL Draft” Icon

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/vgmConfirmation/{id}/update
HTTP Method:	GET
Sending Parameters:	?loadingType=BL
Internal Information:	
View:	loading.loadingPlan.vgmBlConfirmation
View Parameters:	compact('register','select','attachments','usuario','request','blVgm')
Method:	LoadingController@vgmBl
Route Name:	vgmConfirmation
Mandatory Folder:	loading/loadingPlan/*
Controller:	LoadingController

Fields:

- **Submitted BL Draft (*)**: Field Options “Yes/No”
- **Date of Submission (*)**: Required Field, if in the Submitted BL Draft item, the option "Yes" was checked.
- **Time of Submission (*)**: Required Field, if in the Submitted BL Draft item, the option "Yes" was checked.
- **Pre-Draft**: Choose this option only in case of Draft Fictitious. Pre-Draft keeps the task as pending, with a Yellow Status.
- **Attachment Field**: Definition of File type is mandatory before saving.

a. Routine for Submitted BL Draft: Option “Yes”

Once the user has checked the option "Yes", in the field -Submitted BL Draft(*)- The system is prepared for two routes:

By checking the option "Pre-Draft" - The system saves the information that the user has submitted a Pre-Draft and will have to return to the task later to submit the Final Draft.

In this case, the procedural line indicates a pending activity with the color Yellow.

The activity will only be understood as concluded, when the option -Submitted BL Draft(*)- is "Yes", and the checkbox - Pre-Draft- is not selected.

In the Draft option flow submitted as "Yes", the inclusion of date and time of submission becomes mandatory.

b. Routine for Submitted BL Draft: Option "No"

Once the user has checked the option "No", in the field -Submitted BL Draft(*)-

The system is prepared for two routes:

- Transfer or Cancellation

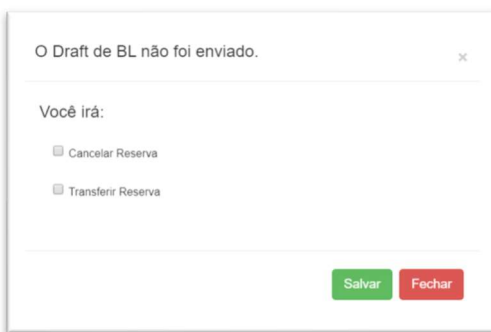


Figure 8 – Screen with option to Cancel or Transfer the BL Draft

If the Booking is Canceled, but there is still the intention to make the shipment with a new Booking, the user, after including the data of the new Booking in the system, can create a "Link" of the previously canceled Booking with the New Booking, and not lose the record of the process history.

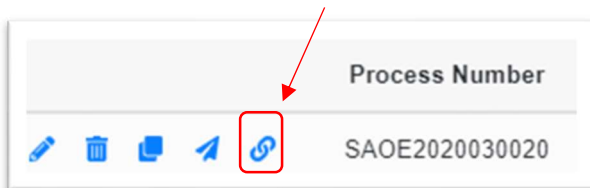


Figure 9 - "Link" Icon

The Link icon above allows the creation of a link between old Canceled Booking to a New Booking.

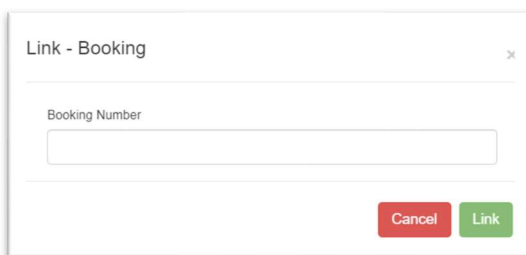


Figure 10 – Link Creation Screen

The editing screen where the New Booking data is available, will present the history with the Booking information, the Old Booking and the Vessel.

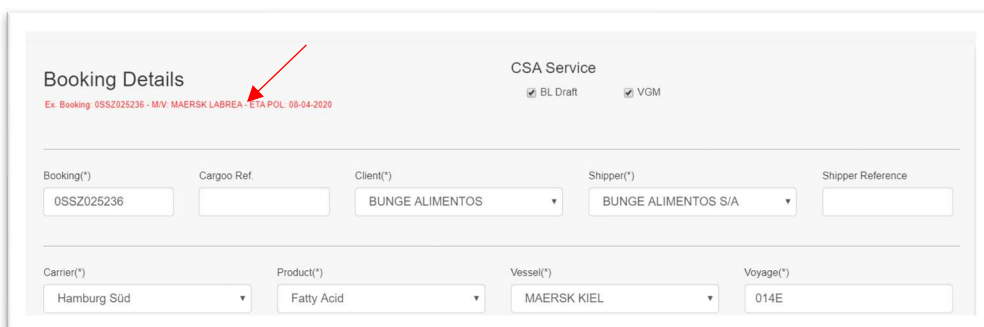


Figure 11 – Screen to edit the New Booking data

****Important:** A Canceled Booking can only be linked to a New Booking once. Therefore, by not sending the Draft, which requires the cancellation of the Booking by the user sent in two copies, one is received with a New Booking with new data for the shipment, or simply cancel the Booking definitely.


In case of a new booking, the same process of "Link" can be performed, to avoid the loss of history.

In case of definite cancellation, the reservation will be viewed in the list of Cancelled Bookings.

Failure to submit the Draft generates the need to transfer the Booking to a new vessel. In this case, the reservation will be forwarded to the "Pending List", where the user can edit the Booking data with the new vessel and save the process.

In this list, the Edit icon becomes available and in the sequence of the saved process, it returns to the "Loading List".

1.7.4 VGM

This action is responsible for updating the VGM sending process. 

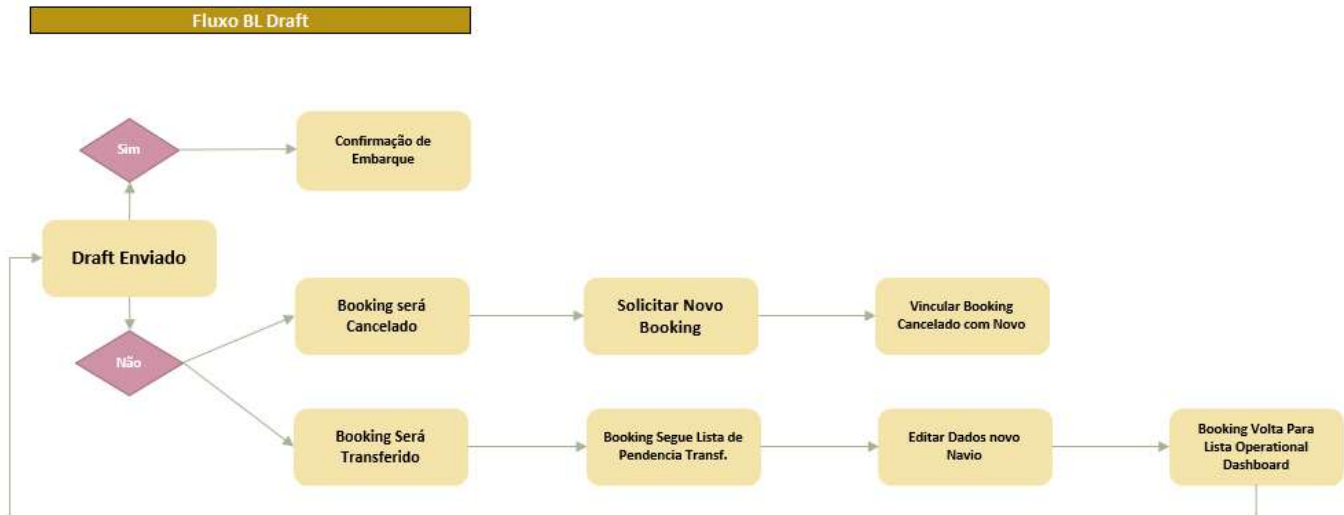


Figure 12 –BL Draft Flow

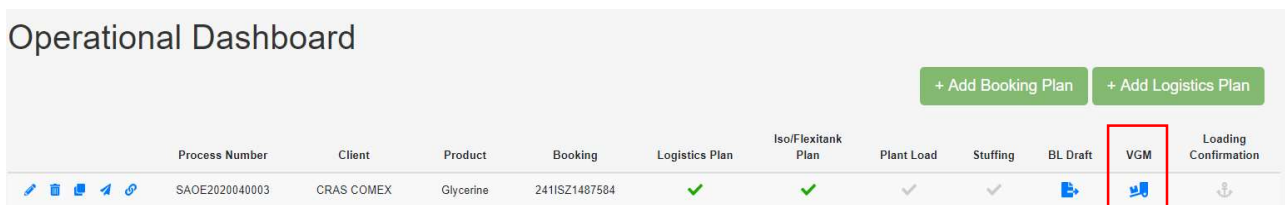


Figure 13 – “VGM” Icon

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/vgmConfirmation/{id}/update
HTTP Method:	GET
Sending Parameters:	?loadingType=VGM
Internal Information:	
View:	loading.loadingPlan.vgmBlConfirmation
View Parameters:	compact('register','select','attachments','usuario','request','blVgm')
Method:	LoadingController@vgmBl
Route Name:	vgmConfirmation
Mandatory Folder:	loading/loadingPlan/*
Controller:	LoadingController

Fields:

- **VGM:** Field Options “Yes/No”
- **Date of Submission (*):** Required Field, if in the VGM, the option “Yes” was checked.
- **Time of Submission (*):** Required Field, if in the VGM Draft, the option “Yes” was checked.

- **Attachment Field:** Definition of File type is mandatory before saving.

VGM Routine: Option "Yes"

After the user has checked the option "Yes", in the – VGM – field:

Figure 14 – VGM Routine Screen

The -Date of Submission- and -Time of Submission(*)- fields become required fields. And the process proceeds to be saved.

VGM Routine: Option "No"

After the user has checked the option "no" in the – VGM – field, the system is ready for 2 routes:

- Transfer or Cancellation

Following the process ed above, where there will always be the option to create "Link" with a New Booking, or definite cancellation, where the Booking will be visible in the list of Cancelled Bookings.

1.7.5 Loading Confirmation (Departured)

This action is responsible for updating the submission process of Loading Confirmation.

Process Number	Client	Product	Booking	ETA POL	Logistics Plan	Iso/Flexitank Plan	Plant Load	Stuffing	BL Draft	VGM	Loading Confirmation
SAOE2020060214	POTENCIAL	GLYCERINE	241ISZ1540564	22-07-2020	✓	✓	✓	✓			📌

Figure 15 – "Anchor" Icon

Figure 16 –Loading Confirmation Screen

The user will be directed to a form to fill in all the data of the loading confirmation.

When the user saves the record, the system changes the status of the process to 'Shipped', which will appear in the 'i' Icon and will be available to start the process in the Financial Module.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	loadingConfirmation/{id}/update
HTTP Method:	GET
Sending Parameters:	?loadingType=BL
Internal Information:	
View:	loading.loadingPlan.loadingConfirmation
View Parameters:	compact('register','select','attachments','usuario','request','bl','blNumber','addBL')
Method:	LoadingController@loadingConfirmation
Route Name:	loadingConfirmation
Mandatory Folder:	loading/loadingPlan/*
Controller:	LoadingController

1.7.5.1 Send an E-mail

After saving the loading confirmation data, the system will open the screen to send the information by email.

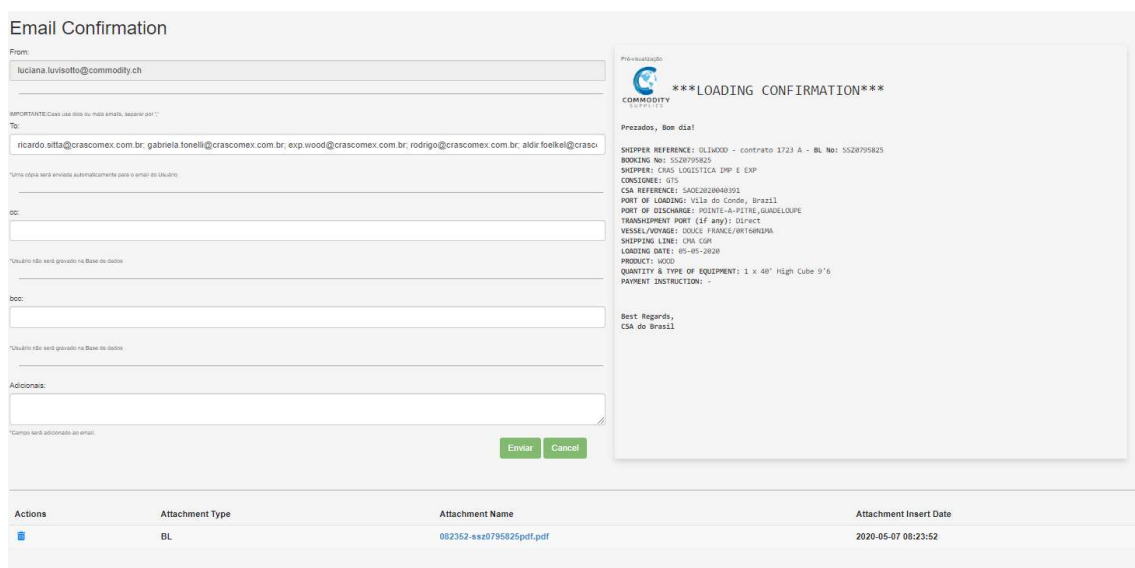


Figure 17 – Email confirmation screen

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/emailView/{id}
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	mail_templates.previouslyEmail
View Parameters:	compact('loading', 'attachments', 'bl')
Method:	MailSendController@previouslyEmail
Mandatory Folder:	mail_templates/*
Route Name:	previouslyEmail
Controller:	MailSendController
Controller (Mail):	App\Mail\BookingConfirmation

1.7.5.2 Email Pre-view

Item	Value
Internal Information:	
View:	mail_templates.mailLoadingConfirmation
View Parameters:	compact('loading', 'attachments', 'loading_confirmation')

Method:	MailSendController@previouslyEmail
Mandatory Folder:	mail_templates/*

1.7.5.3 E-mail Send Controller

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/sendemail/{id}
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
Method:	MailSendController@sendEmail
Mandatory Folder:	mail_templates/*
Route Name:	sendEmail
Controller (Mail):	App\Mail\BookingConfirmation

1.7.6 Cancel

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

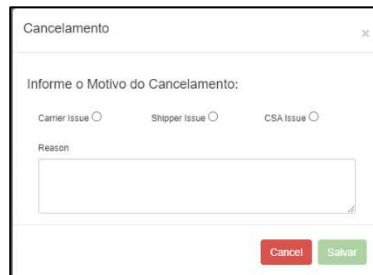
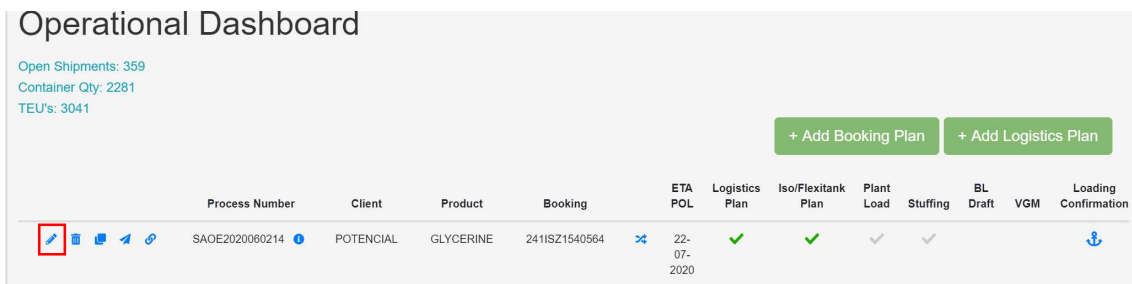


Figure 18 – Cancellation Screen

For the Action to occur, the user must enter the reason for the cancellation.



Process Number	Client	Product	Booking	ETA POL	Logistics Plan	Iso/Flexitank Plan	Plant Load	Stuffing	BL Draft	VGM	Loading Confirmation
SAOE2020060214	POTENCIAL	GLYCERINE	241ISZ1540564	22-07-2020	✓	✓	✓	✓			

Figure 19 – "Pencil" Icon

User must inform the reasons for cancellation.

After this action, the Logistic Programming Register will be available in the menu 0 – Cancelled List.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/loadingPlan/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ include.modal_delete_reason
View Parameters:	data-catid = catid
Method:	LoadingController@destroy
Route Name:	deleteLoading

1.8 Dicionário de Dados Simplificado e Tables essenciais

Identifier	Category		
Name	Loading Request		
Creation Date	08/13/2019	Author	Luciana
Date of last modification	N/A	Author	N/A
Version	1.0	Priority	Essential

UML

Data Dictionary:

Table	loading_requests									
Description	Registration of the entire logistics chain									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
<i>id</i>	bigint(20)	No		x			x			x
users_id	bigint(20)	Yes	NULL	x		x		x		
SAOE	varchar(191)	Yes	NULL	x		x				
shippers_id	bigint(20)	Yes	NULL	x		x		x		
clients_id	bigint(20)	Yes	NULL	x		x		x		
clients_fantasyName	varchar(191)	Yes	NULL	x		x				
shipperReference	varchar(191)	Yes	NULL	x		x				
booking	varchar(191)	Yes	NULL	x		x				
cargooReference	varchar(191)	Yes	NULL			x				
vessel_id	bigint(20)	Yes	NULL	x		x		x		
voyage	varchar(191)	Yes	NULL	x		x				
carriers_id	bigint(20)	Yes	NULL	x		x		x		
platforms_id	bigint(20)	Yes	NULL	x		x		x		
etaPol	date	Yes	NULL	x		x				
etaPod	date	Yes	NULL	x		x				
loading_ports_id	bigint(20)	Yes	NULL	x		x		x		
port_destinations	bigint(20)	Yes	NULL	x		x		x		
draftCutoffDate	date	Yes	NULL	x		x				
draftCutofftime	time	Yes	NULL	x		x				
cargoCutoffDate	date	Yes	NULL	x		x				
cargoCutofftime	time	Yes	NULL	x		x				
vgmCutoffDate	date	Yes	NULL	x		x				
vgmCutoffTime	time	Yes	NULL	x		x				
VGM	tinyint(1)	No	0	x		x				
BLDraft	tinyint(1)	No	0	x		x				
transhipments	bigint(20)	Yes	NULL			x		x		
transitTime	int(11)	Yes	NULL	x		x				
container_types_id	bigint(20)	Yes	NULL	x		x		x		
containerQty	int(11)	Yes	NULL	x		x		x		
loading_terminals_id	bigint(20)	Yes	NULL	x		x		x		

gateInForecast	date	Yes	NULL	x		x				
freeTimeDestination	int(11)	Yes	NULL			x				
freeTimeOrigin	int(11)	Yes	NULL			x				
FreightRate	decimal(12,5)	Yes	NULL			x				
FreightBy	varchar(191)	Yes	NULL			x				
FreightAt	varchar(191)	Yes	NULL			x				
freight_forwarders_id	bigint(20)	Yes	NULL			x		x		
minQtyLoaded	int(11)	Yes	NULL			x				
maxQtyLoaded	int(11)	Yes	NULL			x				
cargo_types_id	bigint(20)	Yes	NULL	x		x		x		
u_n_s_id	bigint(20)	Yes	NULL	0		x		x		
temperature	decimal(12,5)	Yes	NULL	0		x				
foodgrade	enum('Yes', 'No')	Yes	NULL	0		x				
fumigations	enum('Yes', 'No')	Yes	NULL			x				
isotankDetails	text	Yes	NULL			x				
loadingType	text	Yes	NULL	x						
comments	text	Yes	NULL			x				
status	varchar(191)	Yes	Active	x						
email	text	Yes	NULL			x				
hauling_id	bigint(20)	Yes	NULL			x		x		
additionalDetails	text	Yes	NULL			x				
tosubmitted	varchar(191)	Yes	NULL			x				
addcomments	text	Yes	NULL			x				
reasonType	enum('Carrier Issue', 'Shipper Issue', 'CSA Issue')	Yes	NULL			x				
reason	text	Yes	NULL			x				
incoterm	enum('FOB', 'CIF', '')	Yes	NULL	x		x				
imoCutoffDate	date	Yes	NULL	0		x				
imoCutoffTime	time	Yes	NULL	0		x				
inspectionDate	date	Yes	NULL	0		x				
grossweight	decimal(10,3)	Yes	NULL			x				
rateContractNumber	varchar(191)	Yes	NULL			x				
routes_id	bigint(20)	Yes	NULL			x		x		
created_at	timestamp	Yes	NULL							
updated_at	timestamp	Yes	NULL							

EER Diagram:

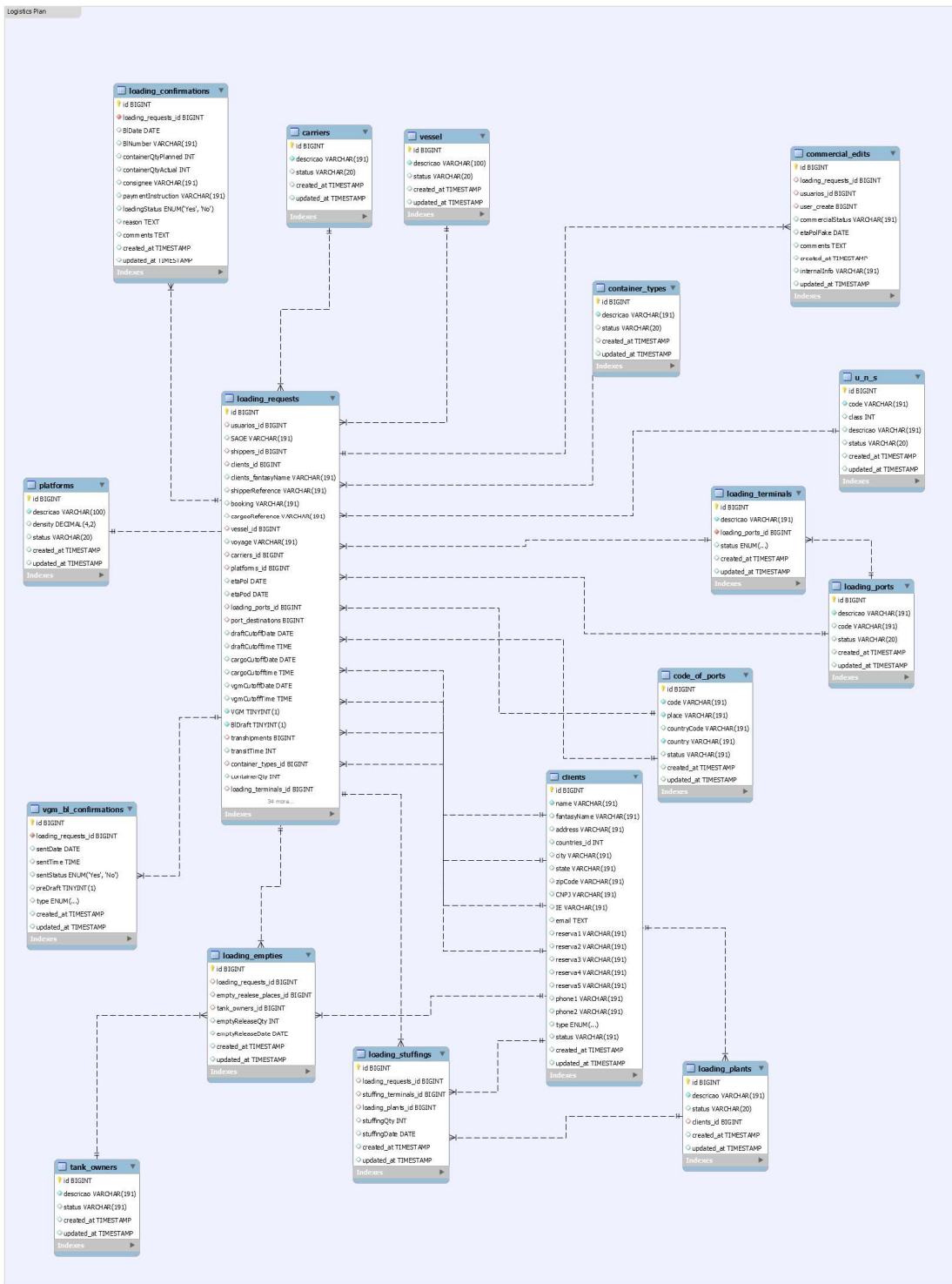


Table: Auxiliares - Log

- email_send_logs
- anexo_types
- loading_links

Views

- effectshipment
- programmedvolume

Validation Rules:

Booking: Required Field. Unique Number.

Cargo Ref.: Unique Number.

Client: Required field. Group by Fantasy Name.

Shipper: According to the relationship clients x shippers

Product: Respecting the relationship clients x Platforms.

ETA POL: Required field. Field imported from Cargo based on the booking confirmation . (rule: >today)

ETA POD: Required field. Field imported from Cargo based on the booking confirmation . (rule: > "Eta POL")

Draft Cutoff Date: Required field. (rule: < "Eta POL")

Cargo Cutoff Date: Required field. (rule: < "Eta POL"; > "Draft Cutoff Date")

VGM Cutoff Date: Required field. (rule: >"Draft Cutoff Date"; <"Cargo Cutoff Date")

Transit Time (days): Full numeric field . Automatic Calculation, Difference in days between ETA POL x ETA POD..

Gate In/Opening Forecast: Required field. (rule: "Eta Pol - 30")

Empty Release Place: Required field, only if **Container Type** = 20' Tank 8'6, if this information is not available, user must select the option "to be informed".

Empty Release Quantity: the "Container Quantity" field of the Booking Details, and the quantity cannot be greater or smaller.

Stuffing Quantity: Required Field, which follows the same validation rule with respect to the "Container Quantity" field in the Booking s, and the quantity cannot be greater or smaller.

Tank Owner: If the field Container type is equal to 20' tank 8'6, the field Tank Owner becomes mandatory, for other options this field is not available.

Cargo Type:

If = "Reefer"; Field "UN" = disabled.

If = "IMO"; Field "Temperature" = disabled.

If = "Regular Cargo"; Fields "UN" + "Temperature" = disabled.

Comments/ Special Instructions: Mandatory field in case of an Update or Transfer type shipment

Container TypeCaso container type = 20' Dry, the fields referring to Isotank will be renamed Flexitank. Comments/ FlexiTank Details, the Tank Owner field will be available to insert data.

Container TypeCaso container type = 20' Tank, the fields referring to Flexitank will be renamed Isotank. Comments/ IsoTank Details, the Tank Owner field will be available to insert data .

loadingType: 2 Values: LGP for Logistics Plan or BKP for Booking Plan.

Status: During the Process in the Operational Dashboard the following statuses are available:

- On Hold – Confirmed Bookings, where the operation is following the entire the logistics programming.
- Migration – Received Bookings, through system integration.
- Pending – If the Booking is transferred or splitted, the reservation is saved as Pending and requires a User Action to return to the operational dashboard.
- Transfer On Hold : In the transfer process the status is activated, only for internal control.
- Transfer e Cancel: For history recording purposes, the bookings that were transferred as well as those that were cancelled are stored .

****Important:** A Canceled Booking can only be linked to a new booking once.

2. Users

Description: Responsible for registering Users in the system.

Preconditions:

- ✓ None

Subsidy to perform the following:

- ✓ Start any task within the TABEX system.

Tipos e níveis de usuários:

Level 1	Related Rights: consultation, editing, registration and exclusion.
Level 2	Related Rights: consultation, editing and registration.
Level 3	Related Rights: consultation and editing.
Level 4	Related Rights: consultation.

Types of Users :

- | | |
|---------------------|-----------------|
| ✓ Freight Forwarder | ✓ Client |
| ✓ Hauling | ✓ Shipper |
| ✓ Stuffing Terminal | ✓ Empty Release |

2.1.1 User List

Consults all system users.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/users
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	users.listar
View Parameters:	compact('usuarios')
Method:	UsuarioController@listar
Route Name:	listarUsuario
Mandatory Folder:	/users

2.1.2 + New

Screen to insert new system users.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/users/registrar
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	users.registrar
View Parameters:	
Method:	UsuarioController@registrar
Route Name:	registrar
Mandatory Folder:	/users

2.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/users/salvar
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	UsuarioController@salvar
Route Name:	salvar
Mandatory Folder:	/users

2.1.3 Edit

Consults all system users.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/users/{id}/editar
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	users.editar
View Parameters:	compact('usuario')
Method:	UsuarioController@editar
Route Name:	editar
Mandatory Folder:	/users

2.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Ponto de extr emidade:	/users/{id}/atualizar
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	UsuarioController@atualizar
Route Name:	atualizar
Mandatory Folder:	/users

2.1.4 Delete

A modal screen is displayed for the confirmation of the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/users/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	UsuarioController@destroy
Route Name:	deletarUsuario
Controller:	UsuarioController

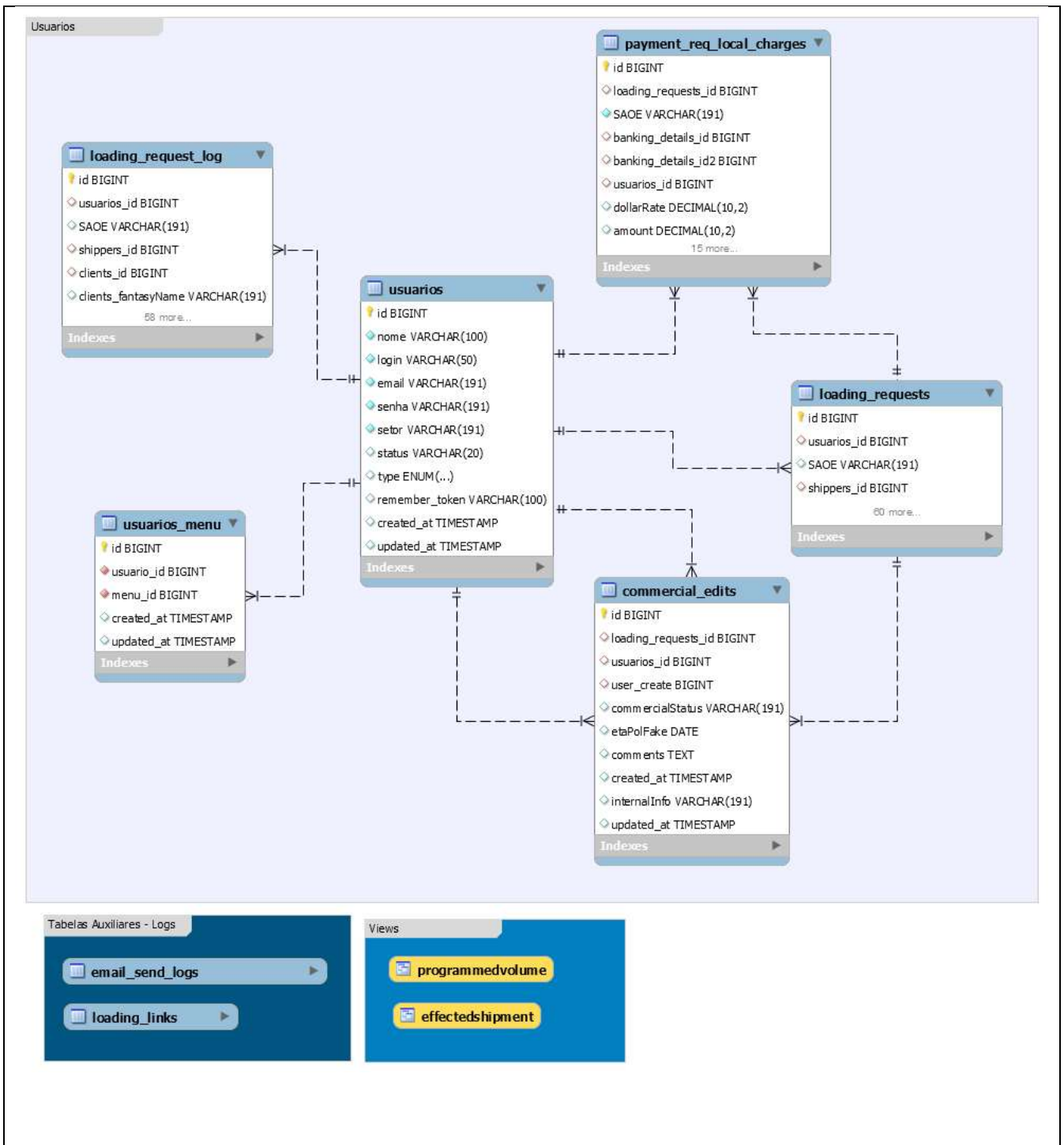
2.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:

Table		users										
Description												
Observations												
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI	ID	
				Insert	Edit							
<i>id</i>	bigint(20)	No		X			x			x		
nome	varchar(100)	No		x	x	X						
login	varchar(50)	No		X	X	X						
email	varchar(191)	No		X	X	X						
senha	varchar(191)	No		X	X	X						
setor	varchar(191)	No		X	X	X						
status	varchar(20)	Yes	0	X	X	X						
type	enum('Freight Forwarder', 'Hauling', 'Client', 'Shipper', 'Stuffing Terminal', 'Empty Release', 'CSA', 'LITTLE DEVIL', 'MASTER DEVIL')	Yes	NULL									
				x	X	X						
remember_token	varchar(100)	Yes	NULL									
created_at	timestamp	Yes	NULL									
updated_at	timestamp	Yes	NULL									

EER Diagram:



3. Menu

Responsible for registering the Menu. Through this Table the system will display the menu containing the appropriate rules;

Preconditions:

- ✓ none

Subsidiary to perform the following:

- ✓ Start any task within the TABEX system.

3.1.1 Menu List

Consult all the existing menus in the system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/menu
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	menu.listar
View Parameters:	compact('menus')
Method:	MenuController@index
Route Name:	listarMenu
Mandatory Folder:	/menu

3.1.2 + New

Screen responsible for inserting new Menu elements.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/menu/create
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	menu.create
View Parameters:	
Method:	MenuController@create
Route Name:	createMenu
Mandatory Folder:	/menu

3.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/menu/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	UsuarioController@store
Route Name:	storeMenu
Mandatory Folder:	/menu

3.1.3 Edit

Responsible for the single edit of the menu record.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/menu/{id}/edit
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	menu.editar
View Parameters:	compact('select', 'menus')
Method:	MenuController@edit
Route Name:	editMenu
Mandatory Folder:	/menu

3.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/users/{id}/update
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
Method:	MenuController@update
Route Name:	updateMenu
Mandatory Folder:	/menu

3.1.4 Delete

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/menu/{id}
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	MenuController@destroy
Route Name:	deleteMenu

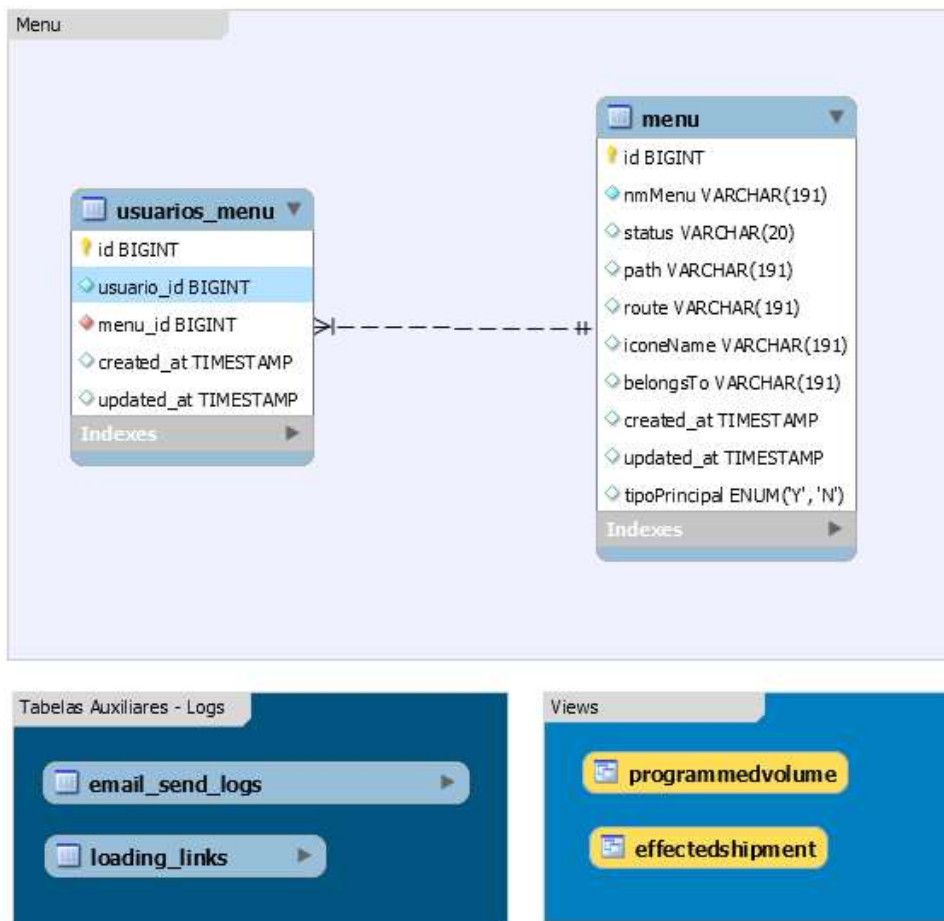
3.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:

Table	Menu									
Description	Registration of the entire logistics chain									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
<i>id</i>	bigint(20)	No		x			x			x
nmMenu	varchar(191)	No		x	x	x				
status	varchar(20)	Yes	0	x	x	x				
path	varchar(191)	Yes	NULL	x	x	x				
route	varchar(191)	Yes	NULL	x	x	x				
iconeName	varchar(191)	Yes	NULL	x	x	x				
belongsTo	varchar(191)	Yes	NULL	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL							
tipoPrincipal	enum('Y', 'N')	Yes	NULL	x	x	x				

EER Diagram:



Rules: In order to properly function, all fields on the table must be filled in according to the programming rule.

4. Users x Menu

Table responsible for controlling the access that each user will be granted within the Tabex system.

In the User Registration or Menu, the administrator must indicate the menu elements available for each user.

Preconditions: Secondary Registration:

- ✓ Menu
- ✓ Users

Subsidy to perform the following:

- ✓ Start any task within the TABEX system

4.1.1 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:

Table	usuarios_menu									
Description	Registration of link of Menu x User									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
<i>id</i>	bigint(20)	No		x			x			x
usuario_id	bigint(20)	No		x	x	x				
menu_id	bigint(20)	No		x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

EER Diagram:

The diagram shows three tables: **usuarios**, **usuarios_menu**, and **menu**. **usuarios** has fields: id BIGINT, nome VARCHAR(100), login VARCHAR(50), email VARCHAR(191), senha VARCHAR(191), setor VARCHAR(191), status VARCHAR(20), type ENUM(...), remember_token VARCHAR(100), created_at TIMESTAMP, updated_at TIMESTAMP. **usuarios_menu** has fields: id BIGINT, usuario_id BIGINT, menu_id BIGINT, created_at TIMESTAMP, updated_at TIMESTAMP. **menu** has fields: id BIGINT, nmMenu VARCHAR(191), status VARCHAR(20), path VARCHAR(191), route VARCHAR(191), iconName VARCHAR(191), belongsTo VARCHAR(191), created_at TIMESTAMP, updated_at TIMESTAMP, tipoPrincipal ENUM('Y', 'N'). Relationships: usuarios (1) to usuarios_menu (many), menu (1) to usuarios_menu (many).

Tabelas Auxiliares - Logs:

- email_send_logs
- loading_links

Views:

- programmedvolume
- effectedsipment

5. Vessel

Domain Table (Master Data) containing the list of vessels available in the Tabex system.

Preconditions: Secondary Registration:

- ✓ User

Subsidy to perform the following:

- ✓ Loading Plan.

5.1.1 Vessel List

Consults all Vessels in the system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/vessel
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	vessel.listar
View Parameters:	compact('vessel')
Method:	VesselController@index
Route Name:	listarVessel
Mandatory Folder:	/vessel

5.1.2 + New

Modal is presented with description field to include a new vessel in the list.

5.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/vessel/salvar
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	VesselController@store
Route Name:	salvarVessel

5.1.3 Edit

Modal is displayed with a description field to edit the selected vessel, in addition to the status button to activate or deactivate the record.

5.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/vessel/{id}/atualizar
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	VesselController@update
Route Name:	atualizarVessel

5.1.4 Delete

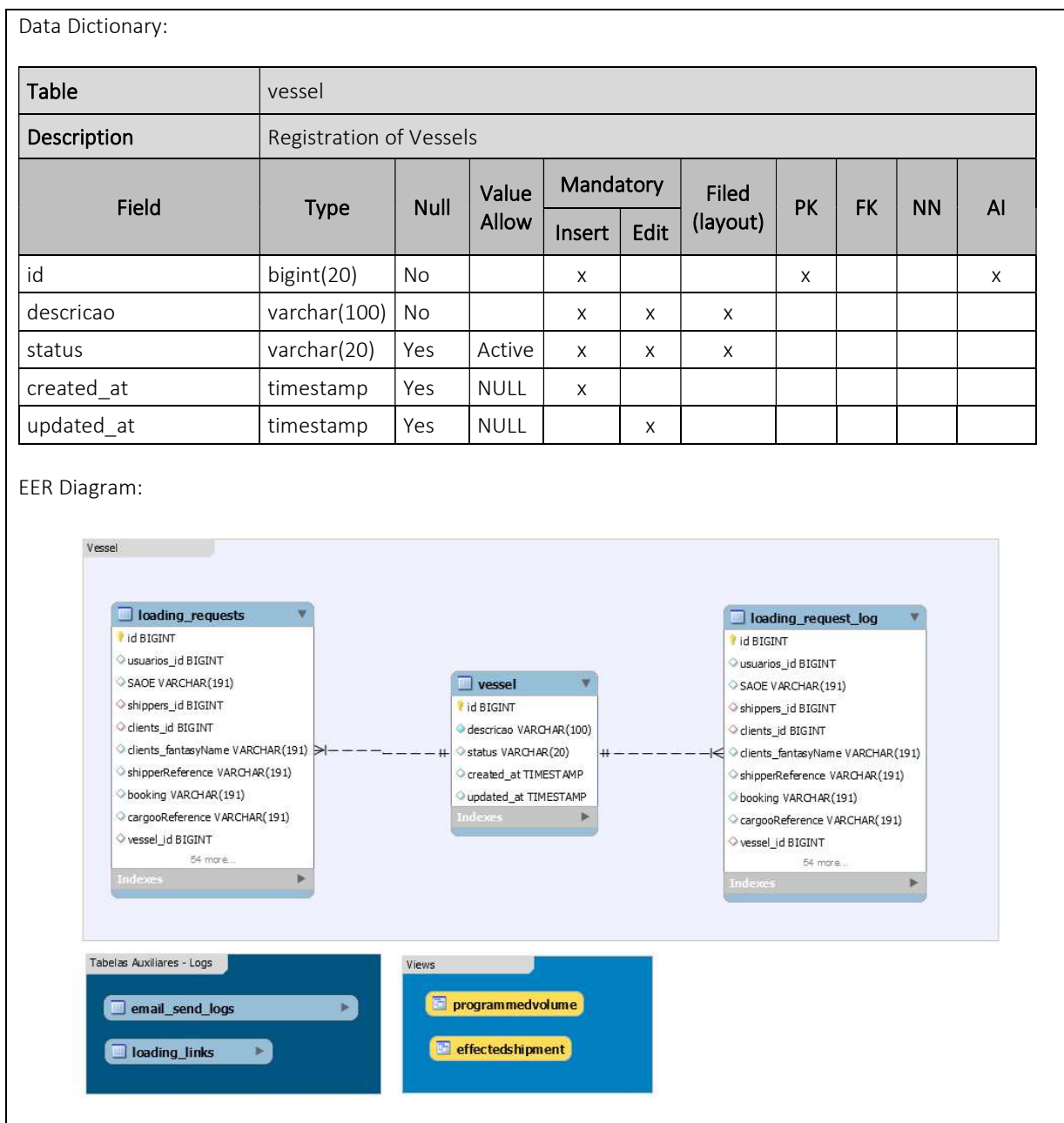
A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/vessel/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	VesselController@destroy
Route Name:	deletarVessel

5.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created:



6. Cargo Types

Domain Table (Master Data) containing the list of cargo types used in the Tabex System.

Preconditions: Secondary Registration:

- ✓ User

Subsidy to perform the following:

- ✓ Loading Plan.

Types of Previously Registered Cargo:

- ✓ IMO
- ✓ Reefer
- ✓ Regular Cargo

6.1.1 Cargo Types List

Consult all Cargo Types in the system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/cargotype
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	cargo.listar
View Parameters:	compact('cargoType')
Method:	CargoTypeController@index
Route Name:	listCargoTypes
Mandatory Folder:	/cargo

6.1.2 + New

Modal is presented with a description field to include a new record in the list.

6.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/cargotype/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	CargoTypeController@store
Route Name:	storeCargoType

6.1.3 Edit

Modal is displayed with a description field to edit the selected record, in addition to the status button to activate or deactivate the record.

6.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/cargotype/{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	CargoTypeController@update
Route Name:	updateCargoType

6.1.4 Delete

A modal screen is displayed to confirm the cancellation of the record.

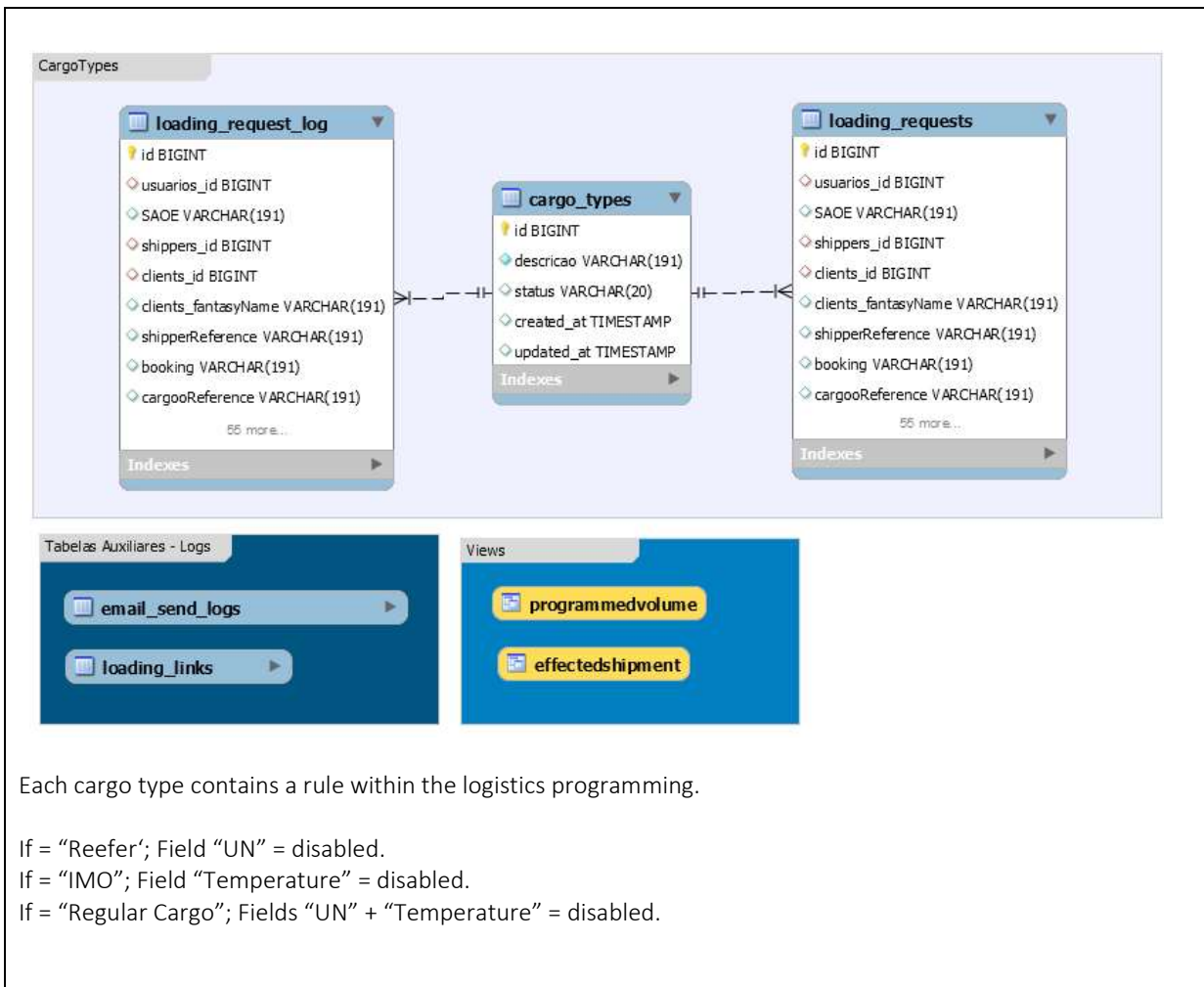
The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/cargotype/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	CargoTypeController@destroy
Route Name:	deleteCargoType

6.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:										
Table	cargo_types									
Description	Registration of Cargo Types									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
<i>id</i>	bigint(20)	No		x			x			x
descricao	varchar(191)	No		x	x	x				
status	varchar(20)	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					
Rules:										



7. Carrier

Domain Table (Master Data) containing the list of carriers used by the system.

Preconditions: Secondary Registration:

- ✓ User

Subsidy to perform the following:

- ✓ Routes
- ✓ Loading Plan.
- ✓ Payment Request
- ✓ Bank Details

7.1.1 List

Consult all types of Carriers in the system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/carrier
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	carrier.listar
View Parameters:	compact('carrier')
Method:	CarrierController@index
Route Name:	listCarriers
Mandatory Folder:	/carrier

7.1.2 + New

Modal is presented with a description field to include a new record in the list.

7.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/carrier/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	CarrierController@store
Route Name:	storeCarrier

7.1.3 Edit

Modal is displayed with a description field to edit the selected record, in addition to the status button to activate or deactivate the record.

7.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/carrier/{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	CarrierController@update
Route Name:	updateCarrier

7.1.4 Delete

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/carrier/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	CarrierController@destroy
Route Name:	deleteCarrier

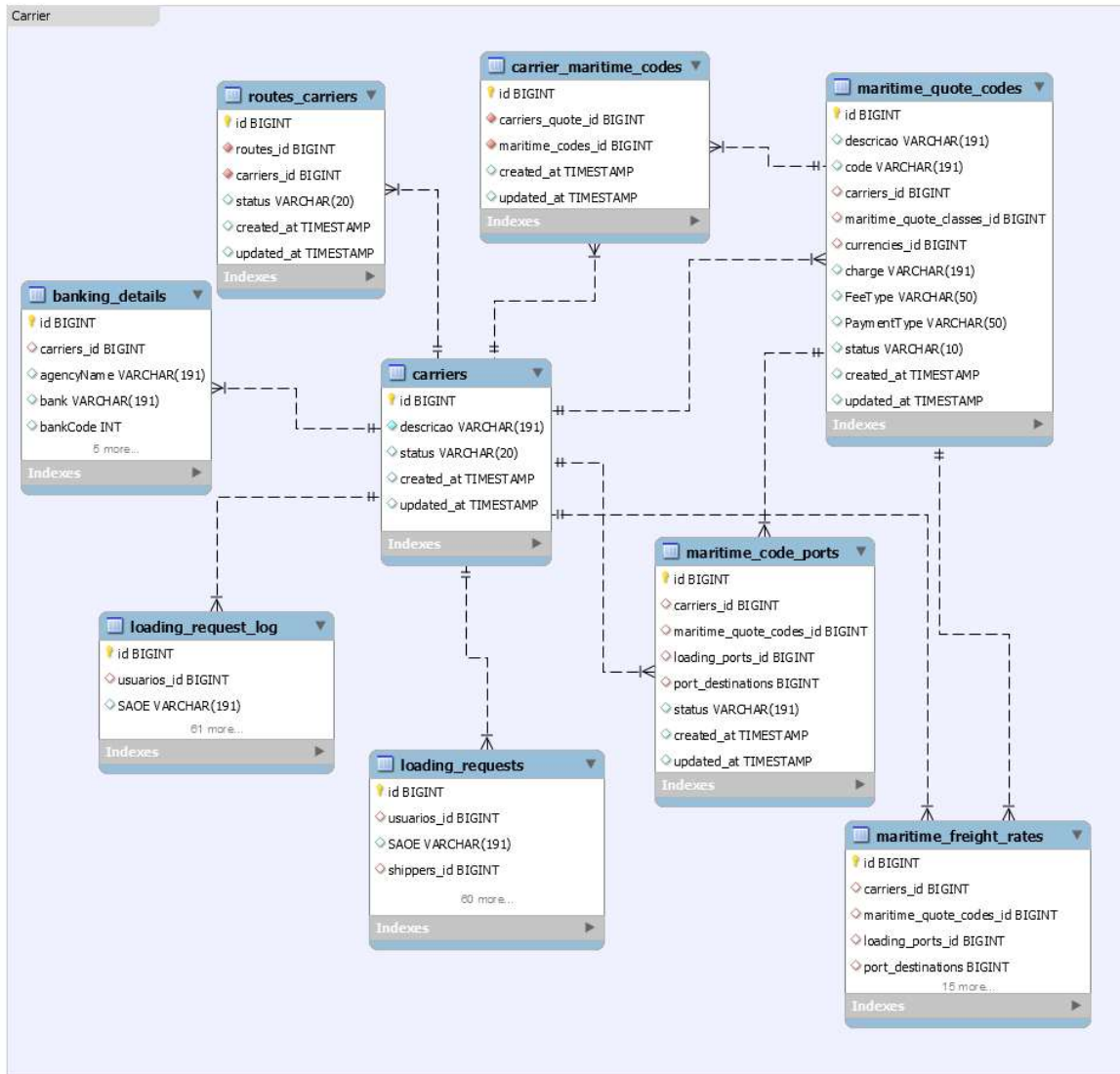
7.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:

Table	carriers									
Description	Registration of Carriers									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
descricao	varchar(191)	No		x	x	x				
status	varchar(20)	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

EER Diagram:



Tabelas Auxiliares - Logs

- email_send_logs
- loading_links

Views

- programmedvolume
- effectedsipment

Rules:

Every Carrier is linked to "N" Routes.

Every Carrier is linked to "N" Bank Accounts.

8. Container Type

Domain Table (Master Data) containing the container types required for logistic programming within the system.

Preconditions: Secondary Registration:

- ✓ User

Subsidy to perform the following:

- ✓ Loading Plan.

Types of Previously Registered Equipment :

- ✓ 1 - 20' Dry 8'6
- ✓ 2 - 40' Dry 8'6
- ✓ 3 - 40' High Cube 9'6
- ✓ 4 - 20' Refrigerated 8'6
- ✓ 5 - 40' Refrigerated 8'6
- ✓ 6 - 40' High Refrigerated 9'6
- ✓ 7 - 20' Tank 8'6
- ✓ 8 - LCL

8.1.1 List

Consult all Equipment Types in the system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/containertype
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	containerType.listar
View Parameters:	compact('containertype')
Method:	ContainerTypeController@index
Route Name:	listContainerTypes
Mandatory Folder:	/containerType

8.1.2 + New

Modal is presented with a description field to include a new record in the list.

8.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/containertype/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	ContainerTypeController@store
Route Name:	storeContainerType

8.1.3 Edit

Modal is displayed with a description field to edit the selected record, in addition to the status button to activate or deactivate the record.

8.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/containertype/{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	ContainerTypeController@update
Route Name:	updateContainerType

8.1.4 Delete

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/containertype/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	ContainerTypeController@destroy
Route Name:	deleteContainerType

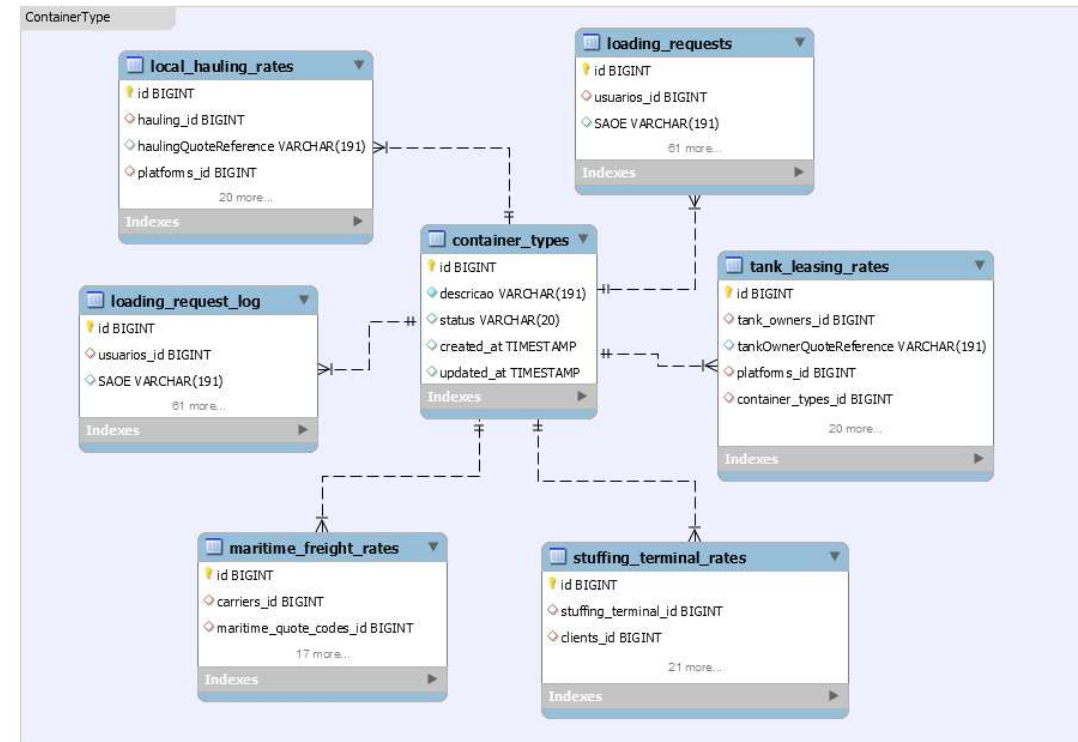
8.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:

Table	container_types									
Description	Registration of Equipment Types (containers)									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
descricao	varchar(191)	No		x	x	x				
status	varchar(20)	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

EER Diagram:



Tabelas Auxiliares - Logs



Views



9. Loading Ports

Domain Table (Master Data) containing the list of Brazilian Ports Previously Registered in the Tabex System.

Preconditions: Secondary Registration:

- ✓ User

Subsidy to perform the following:

- ✓ Loading Plan.

9.1.1 List

Consult all the Brazilian Ports in the system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/loadingport
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	loadingPort.listar
View Parameters:	compact('loadingport')
Method:	LoadingPortController@index
Route Name:	listLoadingPorts
Mandatory Folder:	/loadingPort

9.1.2 + New

Modal is presented with a description field to include a new record in the list.

9.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/loadingport/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	LoadingPortController@store
Route Name:	storeLoadingPort

9.1.3 Edit

Modal is displayed with a description field to edit the selected record, in addition to the status button to activate or deactivate the record.

9.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/loadingport/{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	LoadingPortController@update
Route Name:	updateLoadingPort

9.1.4 Delete

A modal screen is displayed to confirm the cancellation of the record.

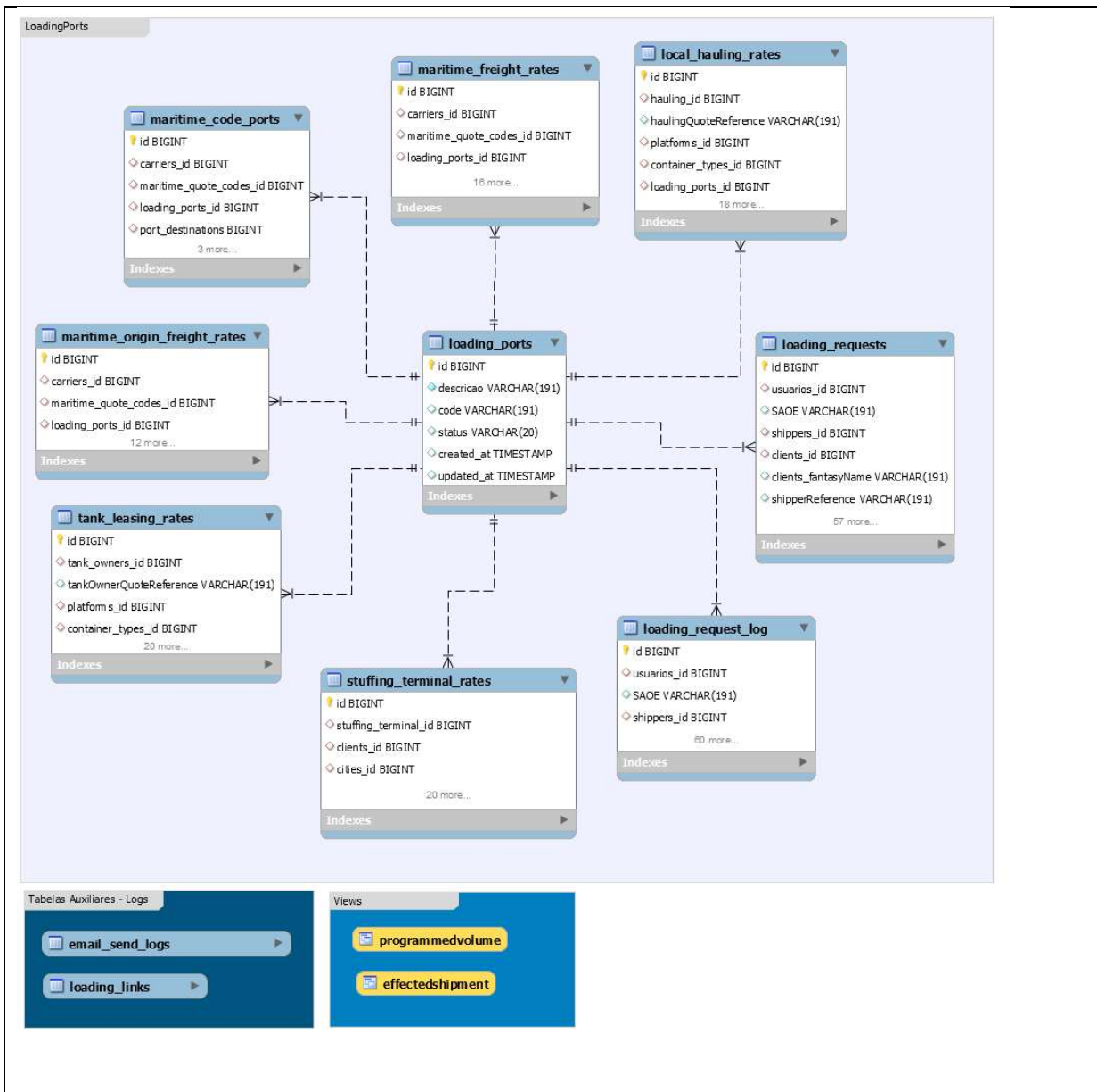
The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/loadingport/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	LoadingPortController@destroy
Route Name:	deleteLoadingPort

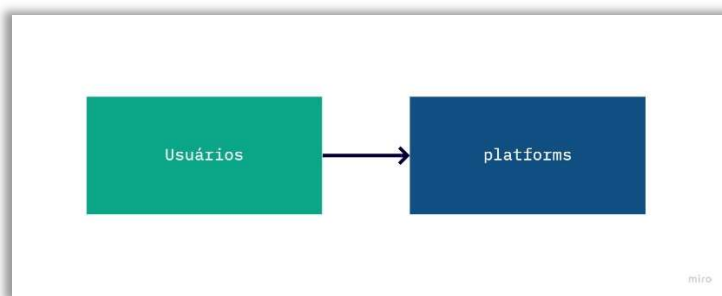
9.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:										
Table	loading_ports									
Description	Registration of Brazilian Ports									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
descricao	varchar(191)	No		x	x	x				
status	varchar(20)	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					
EER Diagram:										



10. Products



Domain Table (Master Data) containing the list of products used in the Tabex system.

Preconditions: Secondary Registration:

- ✓ User

Subsidy to perform the following:

- ✓ Loading Plan
- ✓ Clients

10.1.1 List

Consult all products in the system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/product
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	product.listar
View Parameters:	compact('product')
Method:	ProductController@index
Route Name:	listProducts
Mandatory Folder:	/product

10.1.2 + New

Modal is presented with a description field to include a new record in the list.

10.1.3 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/product/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	ProductController@store
Route Name:	storeProduct

10.1.4 Edit

Modal is displayed with a description field to edit the selected record, in addition to the status button to activate or deactivate the record.

10.1.4.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/product/{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	ProductController@update
Route Name:	updateProduct

10.1.5 Delete

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/product/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	ProductController@destroy
Route Name:	deleteProduct

10.1.6 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:

Table	platforms									
Description	Registration of Products									
Field	Type	Null	Value Allow	Mandatory		Fields (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
descricao	varchar(100)	No		x	x	x				
density	decimal(4,2)	Yes	NULL	x	x	x				
status	varchar(20)	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

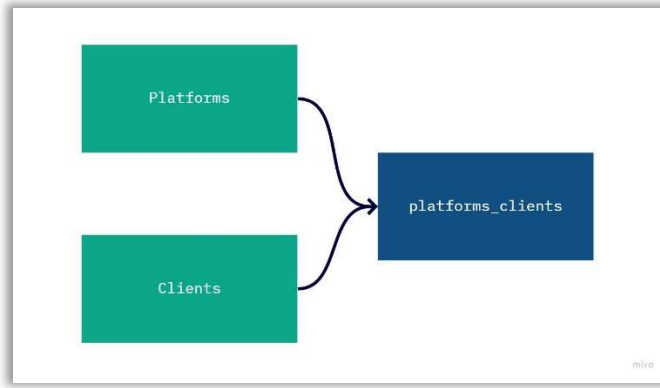
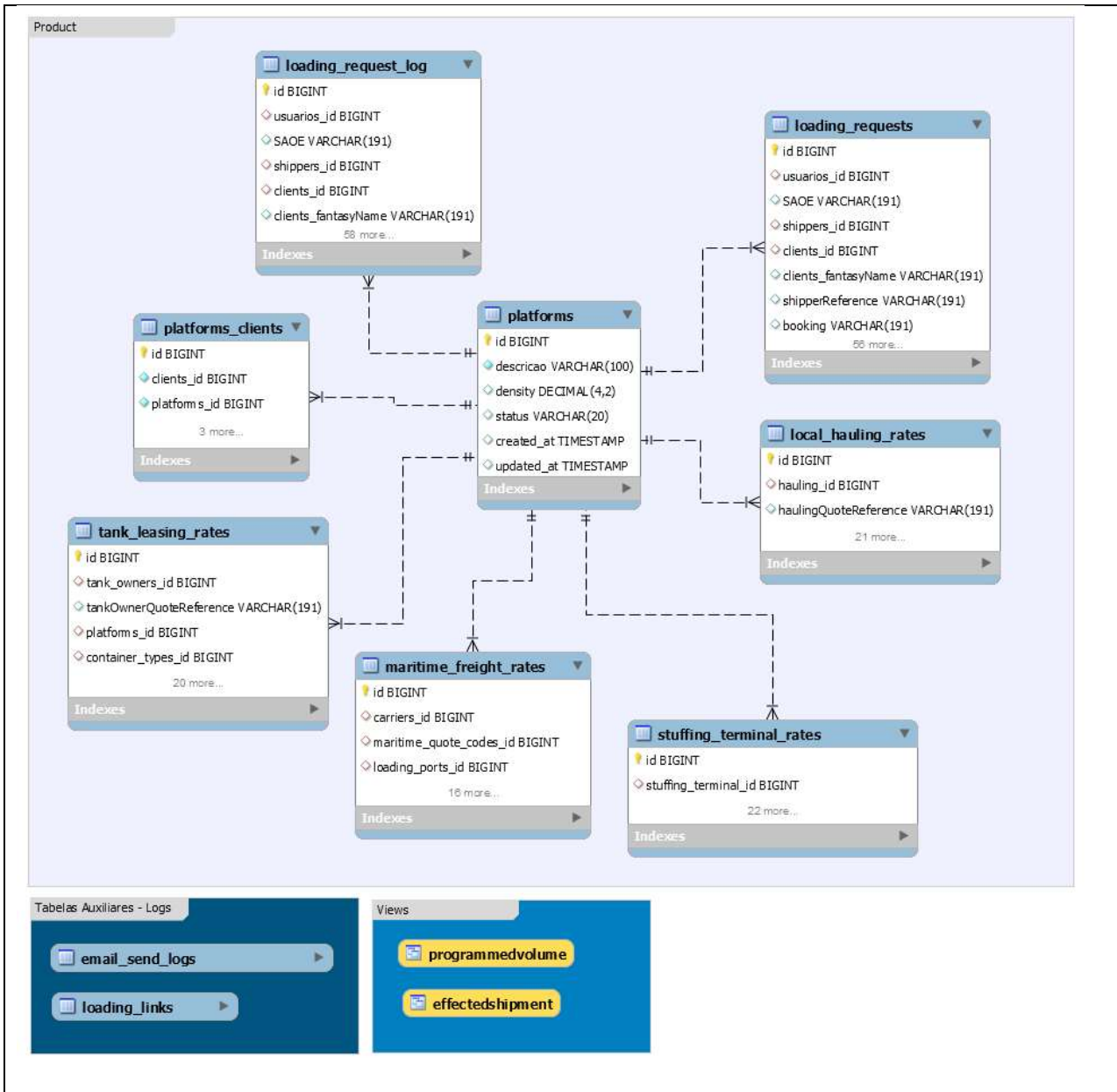
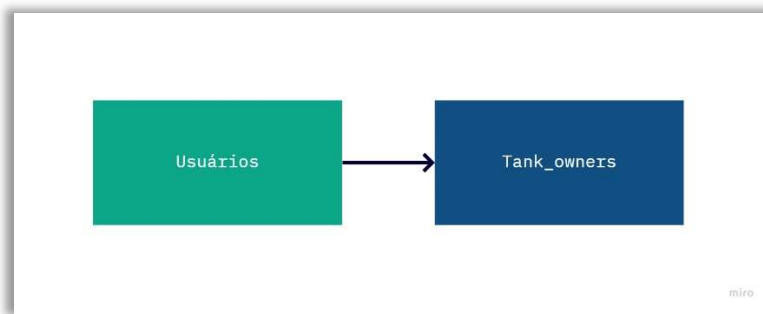


Table		platforms_clients								
Description		Registration of link between client x platforms								
Field	Type	Null	Value Allow	Mandatory		Fields (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
clients_id	bigint(20)	No		x	x	x		x		
platforms_id	bigint(20)	No		x	x	x		x		
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

EER Diagram:



11. Tank Owner



Domain Table (Master Data) containing the list of Tank Owner companies used in the Tabex system.

Preconditions: Secondary Registration:

- ✓ User

Subsidy to perform the following:

- ✓ Loading Plan.

11.1.1 List

Consult all products in the system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/ tankowner
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	tankOwner.listar
View Parameters:	compact(tankowner)
Method:	TankOwnerController@index
Route Name:	listarTankOwners
Mandatory Folder:	/ tankOwner

11.1.2 + New

Modal is presented with a description field to include a new record in the list.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/tankowner/create
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_adicionar.blade.php
View Parameters:	TankOwner
Method:	TankOwnerController @create
Route Name:	createTankOwner
Mandatory Folder:	/ tankOwner

11.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/tankowner/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	TankOwnerController@store'
Route Name:	storeTankOwner

11.1.3 Edit

Modal is displayed with a description field to edit the selected record, in addition to the status button to activate or deactivate the record.

11.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/tankowner/{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	TankOwnerController@update
Route Name:	updateTankOwner

11.1.4 Delete

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/tankowner/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	TankOwnerController@destroy
Route Name:	deleteTankOwner

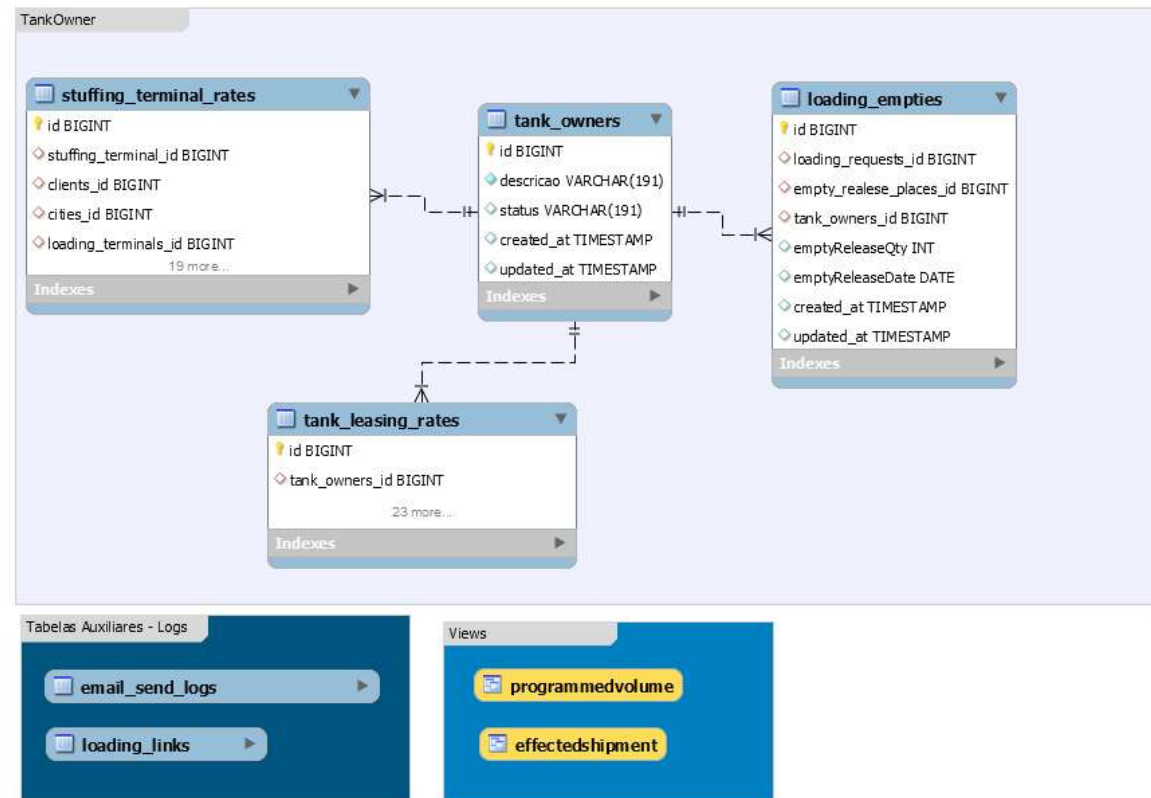
11.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:

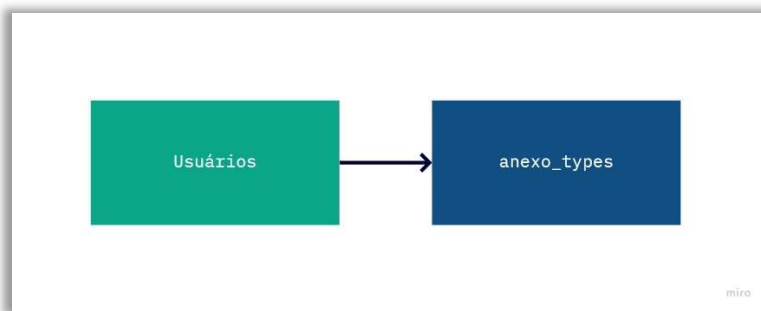
Table		tank_owners								
Description		Registration of Tank Owner companies								
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
descricao	varchar(191)	No		x	x	x				
status	varchar(191)	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

EER Diagram:



* Table under development to include all the necessary data for invoicing.

12. Anexo Types



Domain Table (Master Data) containing the list of Anexo Types used within the Tabex system.

Preconditions: Secondary Registration:

- ✓ User

Subsidy to perform the following:

- ✓ Loading Plan.

12.1.1 List

Consult all Anexo Types in the system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/ anexotype
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	anexoType.listar
View Parameters:	compact(anexoType)
Method:	AnexoTypeController@index
Route Name:	listAnexoType
Mandatory Folder:	/ anexotype

12.1.2 + New

Modal is presented with a description field to include a new record in the list.

12.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/anexotype/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	AnexoTypeController@store
Route Name:	storeAnexoType

12.1.3 Edit

Modal is displayed with a description field to edit the selected record, in addition to the status button to activate or deactivate the record.

12.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/anexotype /{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	AnexoTypeController@update
Route Name:	updateAnexoType

12.1.4 Delete

A modal screen is displayed to confirm the cancellation of the record.

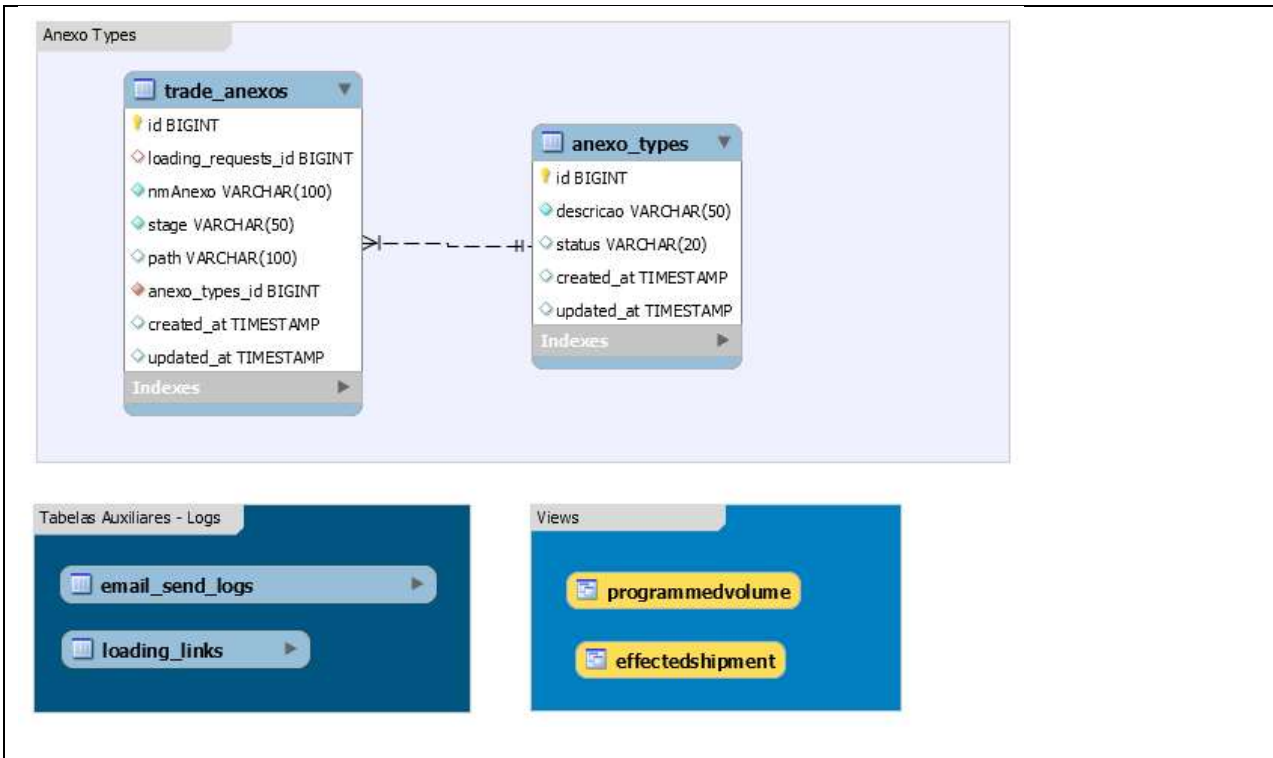
The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/anexotype/{id}
HTTP Method:	GET
Sending Parameters:	AnexoType
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	AnexoTypeController@destroy
Route Name:	deleteAnexoType

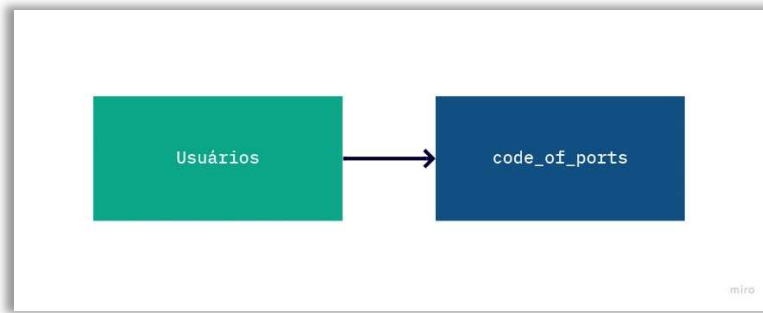
12.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:										
Table	anexo_types									
Description	Registration of Anexo Types in the system									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
descricao	varchar(191)	No		x	x	x				
status	varchar(191)	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					
EER Diagram:										



13. Destination Ports



Domain Table (Master Data) containing the list of Destination and Transshipment Ports used within the Tabex system.

Preconditions: Secondary Registration:

- ✓ User

Subsidy to perform the following:

- ✓ Loading Plan.

13.1.1 List

Consult all ports in the system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/codeofport
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	codeofPort.listar
View Parameters:	compact(code_of_ports)
Method:	CodeOfPortController@index
Route Name:	listCodeOfPort
Mandatory Folder:	/codeofPort

13.1.2 + New

Modal is presented with a description field to include a new record in the list.

13.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/codeofport/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	CodeOfPortController@store
Route Name:	store CodeOfPort

13.1.3 Edit

Modal is displayed with a description field to edit the selected record, in addition to the status button to activate or deactivate the record.

13.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/codeofport /{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	CodeOfPortController@update
Route Name:	updateCodeOfPort

13.1.4 Delete

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

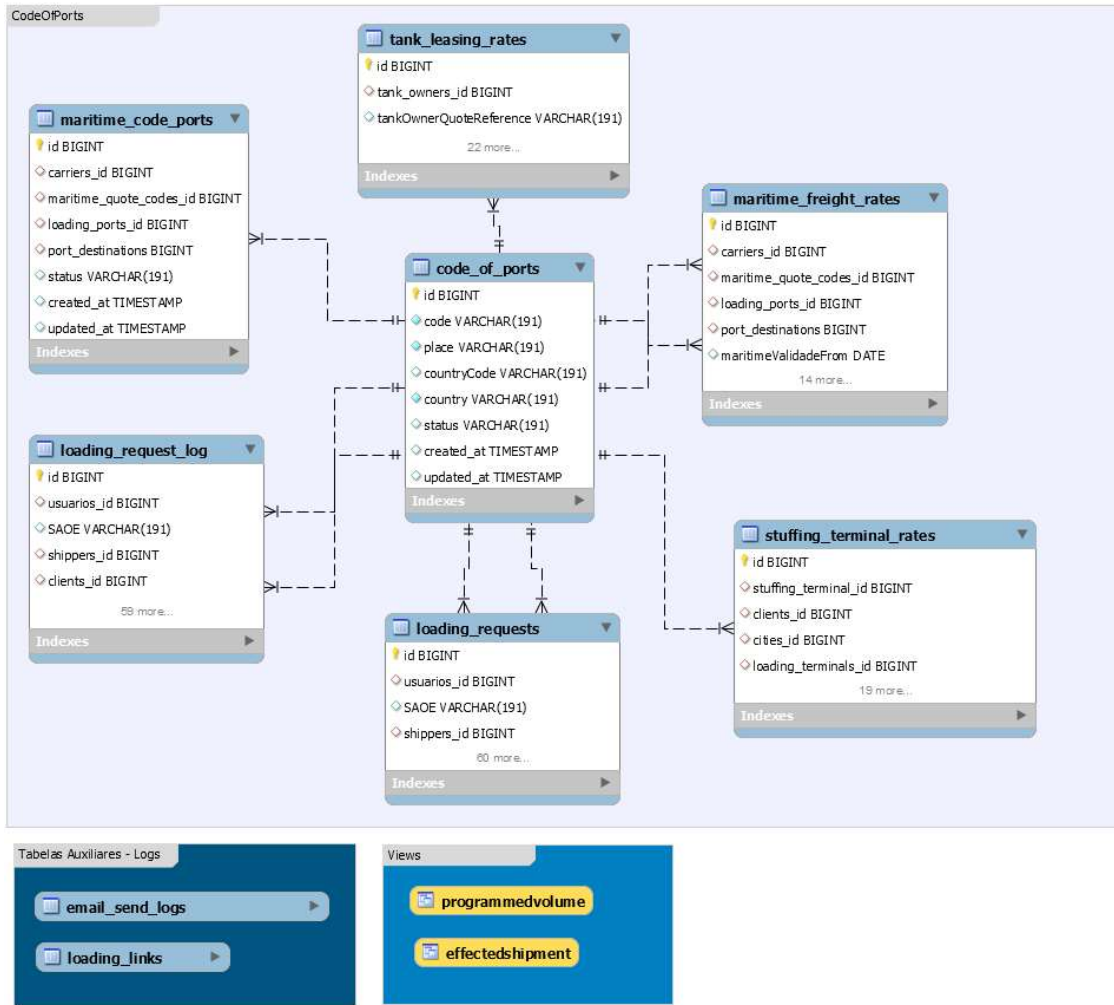
Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/codeofport /{id}
HTTP Method:	GET
Sending Parameters:	AnexoType
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	CodeOfPortController@destroy
Route Name:	deleteCodeOfPort

13.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created: lalalala

Data Dictionary:										
Table	code_of_ports									
Description	Registration of Destination and Transshipment Ports in the System									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		No			x			x
code	varchar(191)	No		x	x	x				
place	varchar(191)	No		x	x	x				
countryCode	varchar(191)	Yes	NULL	x	x	x				
country	varchar(191)	No		x	x	x				
status	varchar(191)	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

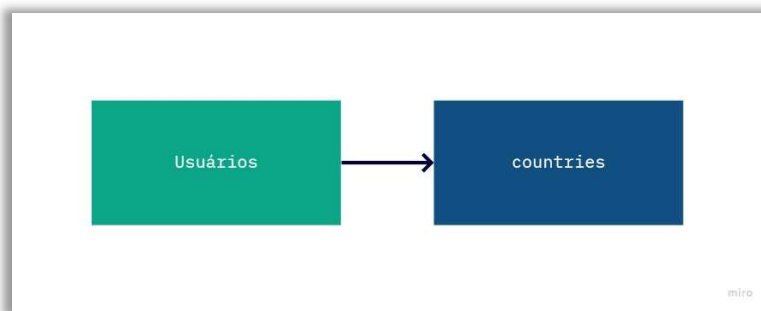
EER Diagram:



Rules:

This table is used for all applications that refer to Destination and Transshipment Ports.

14. Countries



Domain Table (Master Data) containing the list of countries used within the Tabex system.

Preconditions: Secondary Registration:

- ✓ User

Subsidy to perform the following:

- ✓ Clients

14.1.1 List

Consult all countries in the system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/country
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	country.list
View Parameters:	compact(country)
Method:	CountryController@index
Route Name:	listarTankOwners
Mandatory Folder:	/country

14.1.2 + New

Modal is presented with a description field to include a new record in the list.

14.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/country/store
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_adicionar.blade.php
View Parameters:	Country
Method:	CountryController@store
Route Name:	storeCountry

14.1.3 Edit

Modal is displayed with a description field to edit the selected record, in addition to the status button to activate or deactivate the record.

14.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/country/{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	CountryController@update
Route Name:	updateCountry

14.1.4 Delete

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/country/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	CountryController@destroy
Route Name:	deleteCountry

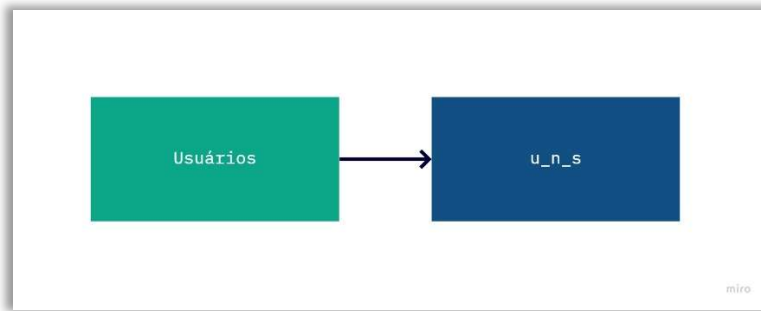
14.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:										
Table	countries									
Description	Registration of all Countries									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
descricao	varchar(191)	No		x	x	x				
status	enum('Ativo', 'Inativo')	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

* Table used for the registration of clients within the system.

15. U.N.



Domain Table (Master Data) containing the list of classes of dangerous products used within the Tabex system.

Preconditions: Secondary Registration:

- ✓ User

Subsidy to perform the following:

- ✓ Loading Plan

15.1.1 List

Consult all classes of dangerous products.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/un
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	UN.listar
View Parameters:	compact(UN)
Method:	UNController@index
Route Name:	listUN
Mandatory Folder:	/UN

15.1.2 + New

Modal is presented with a description field to include a new record in the list.

15.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/un/store
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_adicionar.blade.php
View Parameters:	UN
Method:	UNController@store
Route Name:	storeUN

15.1.3 Edit

Modal is displayed with a description field to edit the selected record, in addition to the status button to activate or deactivate the record.

15.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/un/{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	UNController@update
Route Name:	updateUN

15.1.4 Delete

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

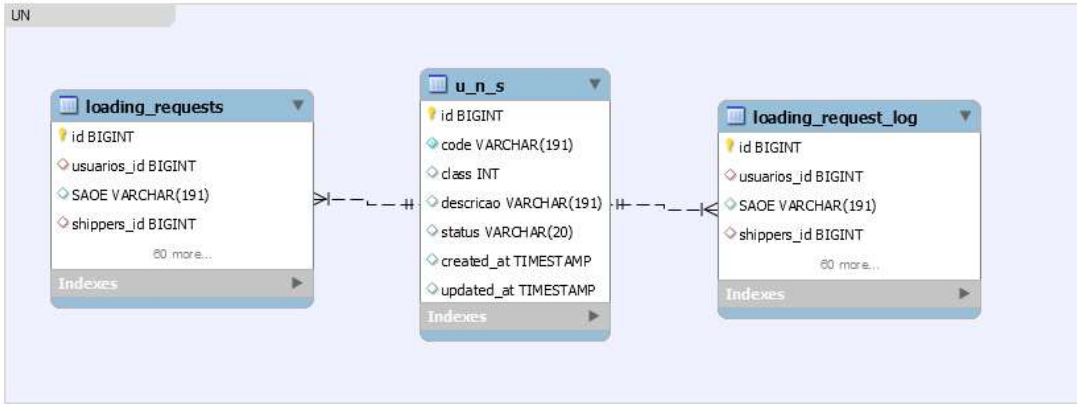
Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/un/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	UNController@destroy
Route Name:	deleteUN

15.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:										
Table	u_n_s									
Description	Registration of classes of dangerous products									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
code	varchar(191)	No		x	x	x				
class	int(11)	Yes	NULL	x	x	x				
descricao	varchar(191)	Yes	NULL	x	x	x				
status	varchar(20)	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

UN:



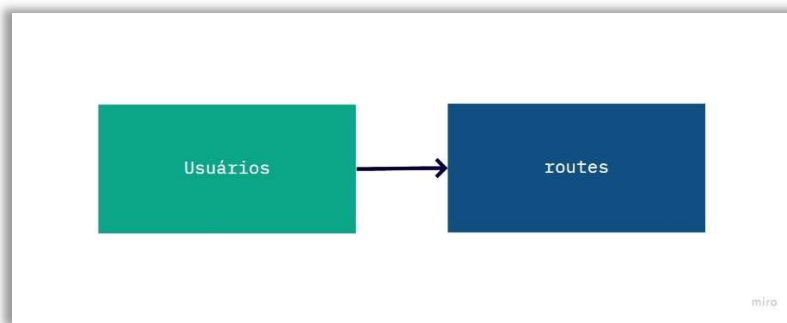
Tabelas Auxiliares - Logs

- email_send_logs
- loading_links

Views

- programedvolume
- effectedsipment

16. Routes



Domain Table (Master Data) containing the list of routes of carriers used within the Tabex system.

Preconditions: Secondary Registration:

- ✓ User
- ✓ Carriers

Subsidy to perform the following:

- ✓ Loading Plan
- ✓ Allocation

16.1.1 List

Consult all registered routes.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/route
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	routes.list
View Parameters:	compact('registers')
Method:	RouteController@index
Route Name:	listRoute
Mandatory Folder:	/route

16.1.2 + New

Modal is presented with a description field to include a new record in the list.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/route/create
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	routes.create
View Parameters:	compact('select')
Method:	RouteController@create
Route Name:	createRoute
Mandatory Folder:	/route

16.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/route/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_adicionar.blade.php
View Parameters:	Route
Method:	RouteController@store
Route Name:	storeRoute

16.1.3 Edit

Modal is displayed with a description field to edit the selected record, in addition to the status button to activate or deactivate the record.

16.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/route/{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	RouteController@update
Route Name:	updateRoute

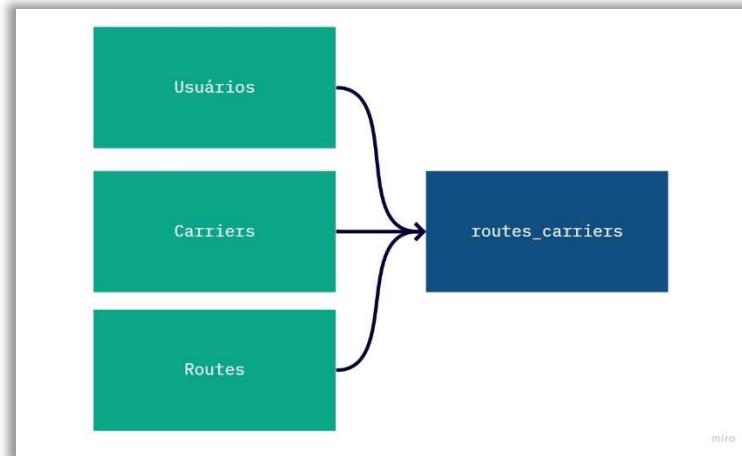
16.1.4 Delete

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/route/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	RouteController@destroy
Route Name:	deleteRoute

16.1.5 UML



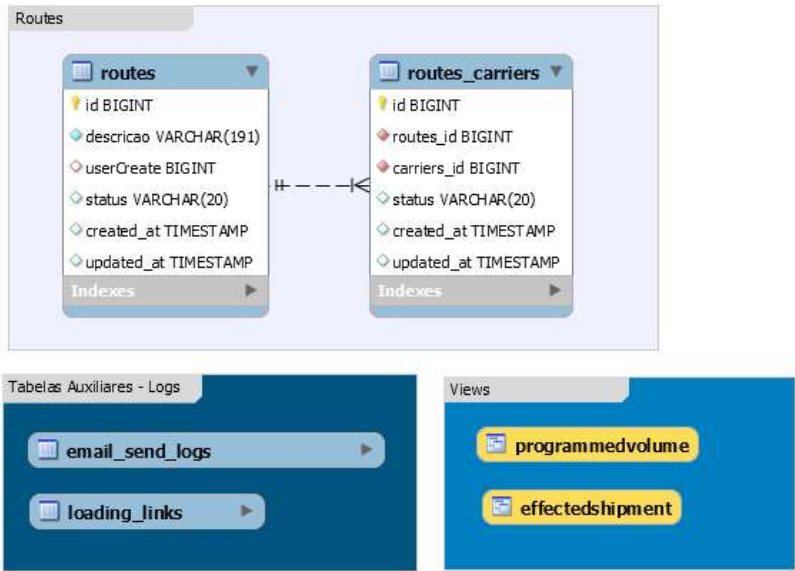
In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:

Table		routes								
Description		Registration of Routes								
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
descricao	varchar(191)	No		x	x	x				
userCreate	bigint(20)	Yes	NULL	x	x	x		x		
status	varchar(20)	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

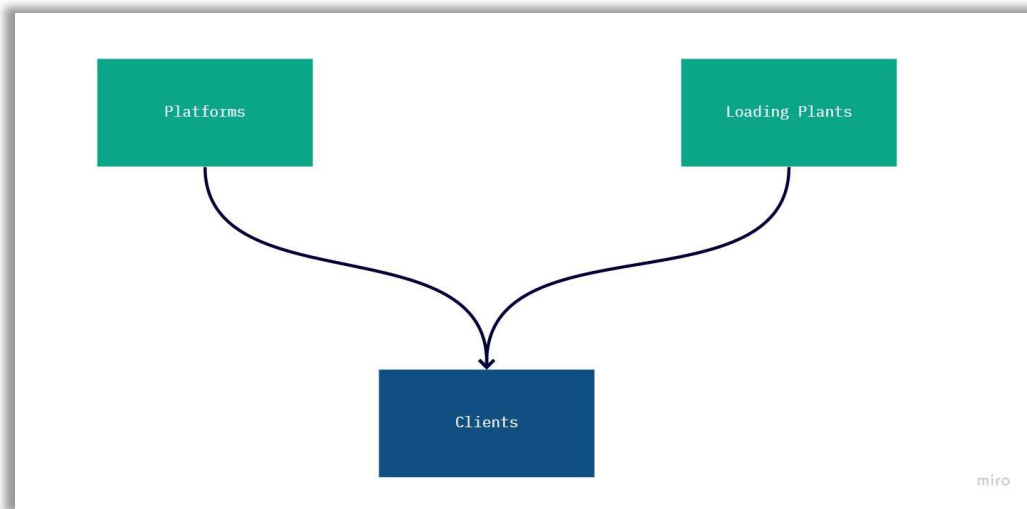
Table		routes_carriers								
Description		Registration of Routes x Carriers								
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
routes_id	bigint(20)	No		x	x	x		x		
carriers_id	bigint(20)	No		x	x	x		x		
status	varchar(20)	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

EER Diagram:



There is the table to create a link between routes and carriers. The registration is made following the database integrity sequence. Route + link.

17. Clients



Application containing all the information from Clients in the Tabex system.

Types of Clients:

- ✓ Freight Forwarder
- ✓ Hauling
- ✓ Shipper
- ✓ Stuffing Terminal
- ✓ Empty Release
- ✓ Flexi Supplier

Preconditions: Secondary Registration:

- ✓ Users
- ✓ Platforms (Product)
- ✓ Loading Plan

Subsidy to perform the following:

- ✓ Start any task within the TABEX system

17.1.1 Companies List

Consult all records within the Table Clients in the system, regardless of Type.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/client/list
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Client.list
View Parameters:	compact('clientes')
Method:	ClientController@index
Route Name:	listClients
Mandatory Folder:	/Client

17.1.1.1 Lists separated by Type

Item	Value
Type:	Client
Endpoint:	client/list/client
HTTP Method:	GET
Method:	ClientController@index
Route Name:	listClientClient

Item	Value
Type:	Hauling
Endpoint:	client/list/hauling
HTTP Method:	GET
Method:	ClientController@indexHauling
Route Name:	listClientHauling

Item	Value
Type:	Freight Fowarder
Endpoint:	client/list/freightforwarder
HTTP Method:	GET
Method:	ClientController@indexFreightForwarder
Route Name:	listClientFreightForwarder

Item	Value
Type:	Shipper
Endpoint:	client/list/shipper
HTTP Method:	GET
Method:	ClientController@indexShipper
Route Name:	listClientShipper

Item	Value
Type:	Stuffing Terminal
Endpoint:	client/list/stuffingterminal
HTTP Method:	GET
Method:	ClientController@indexStuffingTerminal
Route Name:	listClientStuffingTerminal

Item	Value
Type:	Empty Release
Endpoint:	client/list/emptyrelease
HTTP Method:	GET
Method:	ClientController@indexEmptyRelease
Route Name:	listClientEmptyRelease

Item	Value
Type:	Stuffing Terminal
Endpoint:	client/list/flexisupplier
HTTP Method:	GET
Method:	ClientController@indexFlexiSupplier
Route Name:	listClientFlexiSupplier

17.1.2 + New

Screen with form to register a new Client.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/client/create
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	ClientController@create
Route Name:	createClient

17.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/ client /store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	ClientController @store
Route Name:	storeClient

17.1.3 Edit

Screen responsible for editing the record.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/client/{id}/edit
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	ClientController@edit
Route Name:	editClient

17.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/client/{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	ClientController@update
Route Name:	updateClient

17.1.4 Delete

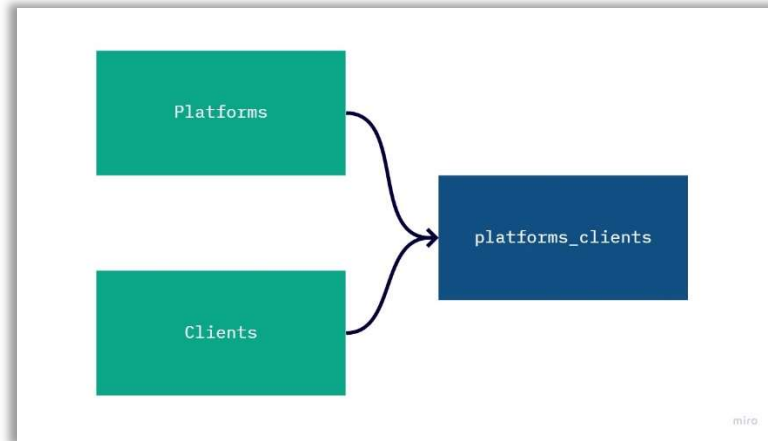
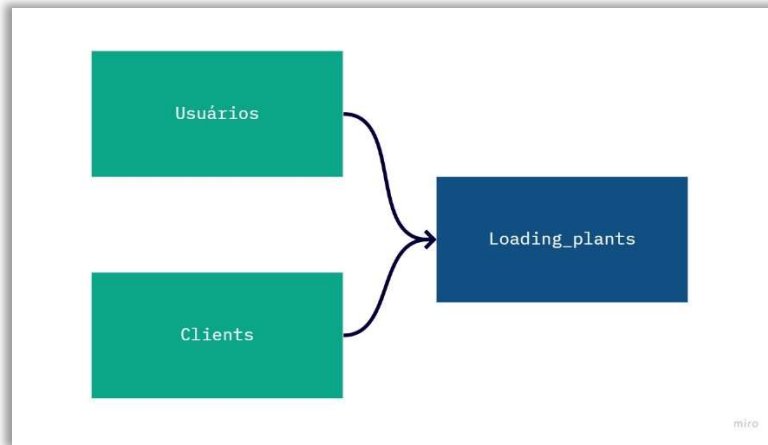
A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/client/{id}
HTTP Method:	GET
Sending Parameters:	null

Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	ClientController@destroy
Route Name:	deleteClient

17.1.5 UML



In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:

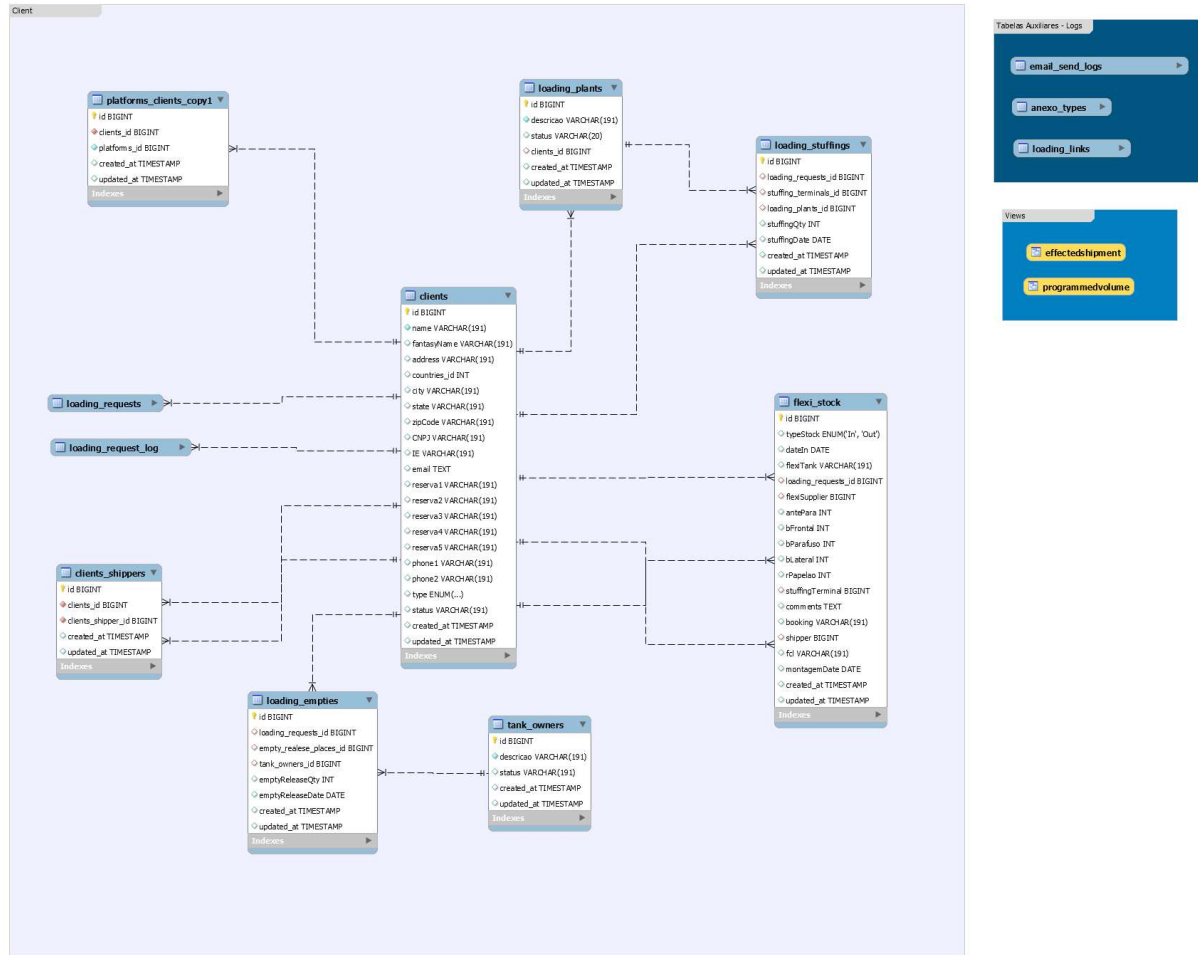
clien		Clients								
Description		Registration of the entire logistics chain								
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
name	varchar(191)	No		x	x	x				
fantasyName	varchar(191)	Yes	NULL	x	x	x				
address	varchar(191)	Yes	NULL	x	x	x				
countries_id	int(11)	Yes	16	x	x	x				
city	varchar(191)	Yes	NULL	x	x	x				
state	varchar(191)	Yes	NULL	x	x	x				
zipCode	varchar(191)	Yes	NULL	x	x	x				
CNPJ	varchar(191)	Yes	NULL	x	x	x				
IE	varchar(191)	Yes	NULL	x	x	x				

email	text	Yes	NULL	x	x	x				
reserva1	varchar(191)	Yes	NULL	x	x	x				
reserva2	varchar(191)	Yes	NULL							
reserva3	varchar(191)	Yes	NULL							
reserva4	varchar(191)	Yes	NULL							
reserva5	varchar(191)	Yes	NULL							
phone1	varchar(191)	Yes	NULL							
phone2	varchar(191)	Yes	NULL	x	x	x				
type	enum('Freight Forwarder', 'Hauling', 'Client', 'Shipper', 'Stuffing Terminal', 'Empty Release', 'Flexi Supplier')	Yes	NULL	x	x	x				
status	varchar(191)	Yes	Active	x	x	x				
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

Table	loading_plants									
Description	Registration of plants according to each client									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
descricao	varchar(191)	No		x	x	x				
status	varchar(20)	Yes	Active	x	x	x				
clients_id	bigint(20)	Yes	NULL	x	x	x		x		
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

Table	platforms_clients									
Description	Registration of link between client x platforms									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
clients_id	bigint(20)	No		x	x	x		x		
platforms_id	bigint(20)	No		x	x	x		x		
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

EER Diagram:



Rules:

If the registration is of a Brazilian Company, it is mandatory to fill out the CEP, CNPJ and IE.

Every Client represents a company within the system, whether a client or supplier.

Each Plant (supplier) belongs to a Client.

If Client Type = 'client' the Loading Plant link with the Client is enabled. In case there is no link, the record is created automatically. Save the link and plant respecting the database integrity. Client id + Loading Plant.

If Client Type = 'client' the Product (platforms) link with the Client is enabled. Unlike Loading Plant, the product must be previously registered in the system.

18. Banking Details

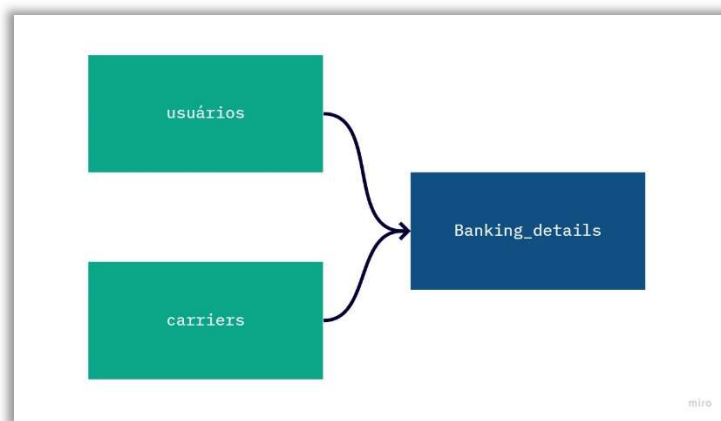


Table containing the list of Banking Details of each Carrier used within the Tabex system.

Preconditions: Secondary Registration:

- ✓ User
- ✓ Carriers

Subsidy to perform the following:

- ✓ Payment Request
- ✓ Finance

18.1.1 List

Consult all Banking Details registered in the system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/banking-details
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	bankDetails.list
View Parameters:	compact('registers','select')
Method:	BankingDetailController@index
Route Name:	listBankDetail
Mandatory Folder:	/bankDetails

18.1.2 + New

Modal is presented with a description field to include a new record in the list.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/banking-details/create
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	bankDetails.create
View Parameters:	compact('select')
Method:	BankingDetailController@create
Route Name:	createBankDetail
Mandatory Folder:	/bankDetails

18.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/banking-details/store
HTTP Method:	POST
Sending Parameters:	null
Internal Information:	
Method:	BankingDetailController@store
Route Name:	storeBankDetail

18.1.3 Edit

Form is presented with description field to edit the selected record, in addition to the status button to activate or deactivate the record.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/banking-details/{id}/edit
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	bankDetails.edit
View Parameters:	compact('select','registers')
Method:	BankingDetailController@edit
Route Name:	editBankDetail
Mandatory Folder:	/ bankDetails

18.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/banking-details/{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	BankingDetailController@update
Route Name:	updateBankDetail

18.1.4 Delete

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/banking-details/{id}/delete
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	BankingDetailController@destroy
Route Name:	deleteBankDetail

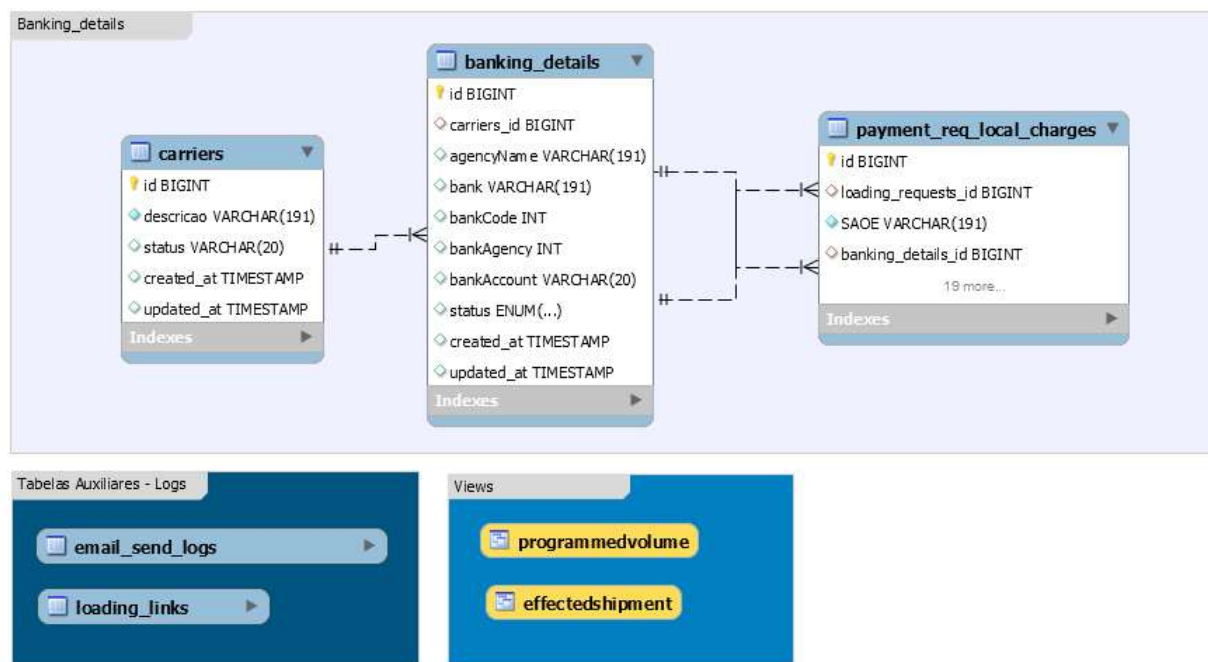
18.1.5 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:

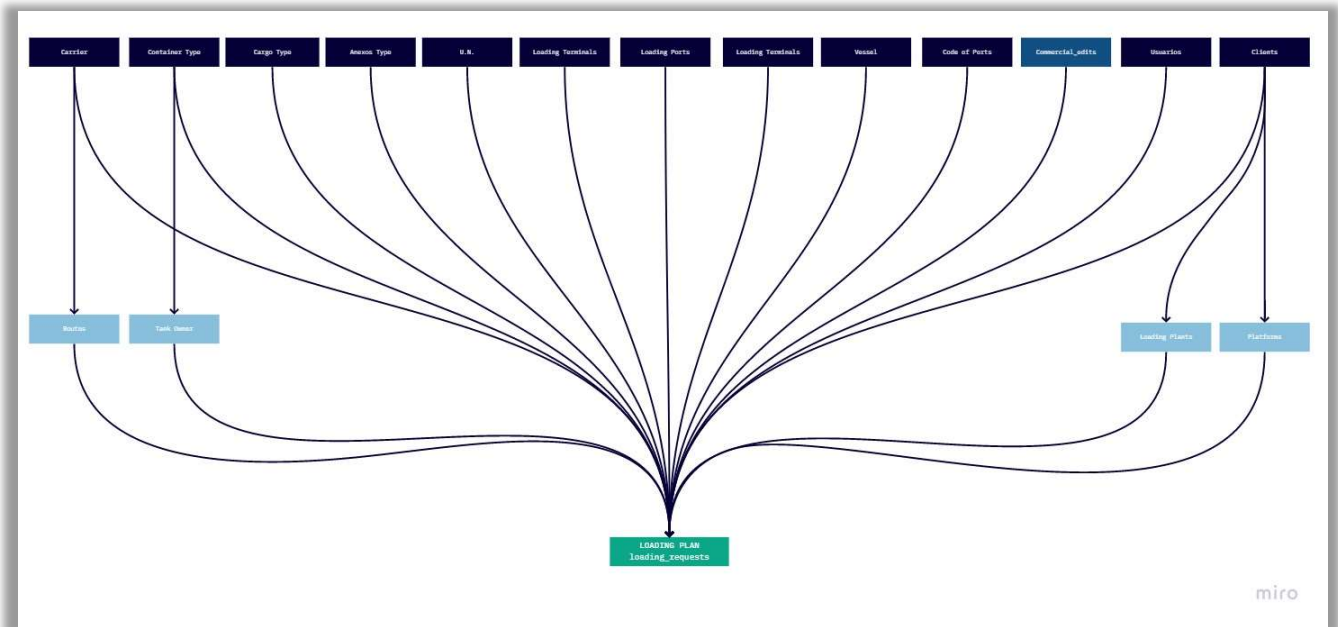
Table	banking_details									
Description	Registration of Banking Accounts									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
carriers_id	bigint(20)	Yes	NULL	x	x	x	x			
agencyName	varchar(191)	Yes	NULL	x	x	x				
bank	varchar(191)	Yes	NULL	x	x	x				
bankCode	int(11)	Yes	NULL	x	x	x				
bankAgency	int(11)	Yes	NULL	x	x	x				
bankAccount	varchar(20)	Yes	NULL	x	x	x				
status	enum('Ativo', 'Inativo')	Yes	Active	x						
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

EER Diagram:



There is a table to create a link between bank and carriers. The registration is made following the database integrity sequence.

19. Allocation



Represents an application with all the bookings/logistics programming currently followed or the pre-bookings available for future services.

Application with replica of Table Loading_requests. All methods and links refer to the Loading Plan, with the exception of commercial_edits

Preconditions: Secondary Registration:

- ✓ Shippers (Client)
- ✓ Loading Terminal (Client)
- ✓ Tank Owner (Client)
- ✓ Empty Release Place (Client)
- ✓ Stuffing Place (Client)
- ✓ Freight/Forwarder (Client)
- ✓ Commercial Edit
- ✓ Hauling (Client)
- ✓ Client
- ✓ Carriers
- ✓ Product
- ✓ Vessel
- ✓ Loading Port (Brazil)
- ✓ Code of Ports
- ✓ Container Type
- ✓ Cargo Type
- ✓ U.N.
- ✓ Anexos Type
- ✓ Routes

Subsidy to perform the following:

- ✓ Loading Plan
- ✓ VGM
- ✓ Draft BL
- ✓ Loading Confirmation (departed)

19.1.1 Allocation List

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/allocationPosition/list
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	allocation.list
View Parameters:	compact('registers','select', 'returnSearchSelect')
Method:	PreBookingController@index
Route Name:	allocationPosition
Mandatory Folder:	/allocation

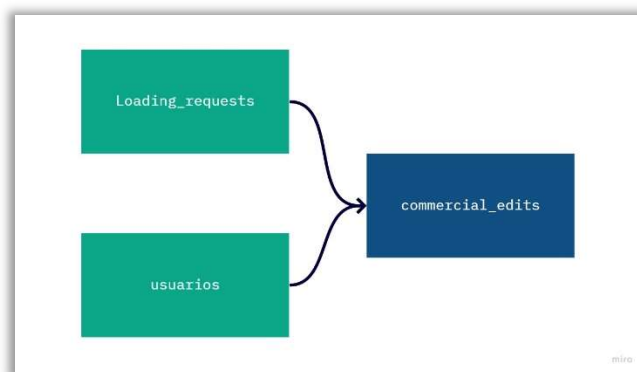
19.1.2 Commercial Edit Modal

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	allocationPosition/commercialEdit
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Modal -> allocation/allocation.modal_commercial_edit.blade.php
View Parameters:	data-idcat = idcat
Method:	PreBookingController@commercialEdit
Route Name:	commercialEdit

19.1.3 Commercial Edit Unlock – Stand By Status

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/allocationPosition/commercialEdit/{id}
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	Modal -> include /include.blade.php / CancelStandByStatus
View Parameters:	data-catid= catid
Method:	PreBookingController@CancelStandByStatus
Route Name:	commercialEdit

19.1.4 UML




In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:


Table	commercial_edits									
Description	Registration of tasks to be executed or delegated to the Operational Area									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No			x		x			x
loading_requests_id	bigint(20)	Yes	NULL			x		x		
users_id	bigint(20)	Yes	NULL			x		x		
user_create	bigint(20)	Yes	NULL			x		x		
commercialStatus	varchar(191)	Yes	Status Ok			x				
etaPolFake	date	Yes	NULL			x				
comments	text	Yes	NULL			x				
created_at	timestamp	Yes	NULL	x						
internalInfo	varchar(191)	Yes	NULL							
updated_at	timestamp	Yes	NULL		x					

Rules:

Commercial Edits  is an action guide for the Operations Team. Stores the history of each requested alteration and only makes available the last requested action in the application.

Available Status: Stand By, Booking to Be Cancelled, Booking to be transferred, Waiting for transference, Waiting for cancelling, Cancelled, Transferred, Submitted.

Actions: Change ETA – provide the desired ETA. Direct the Booking to a Specific User.

Stand By Status: If the status is saved as Stand By, the record will change colors to Yellow, and will display an icon to prevent any changes until the unlock button is activated. 


Pre-status of Booking (condition and not field):

- Fake = Client e Shipper as (=) CSA
- Pré Booking = Client equal (=) to CSA and Booking directed to a Client.
- Real = Booking with different Client and Shipper (<>) from CSA.

Color Rules:

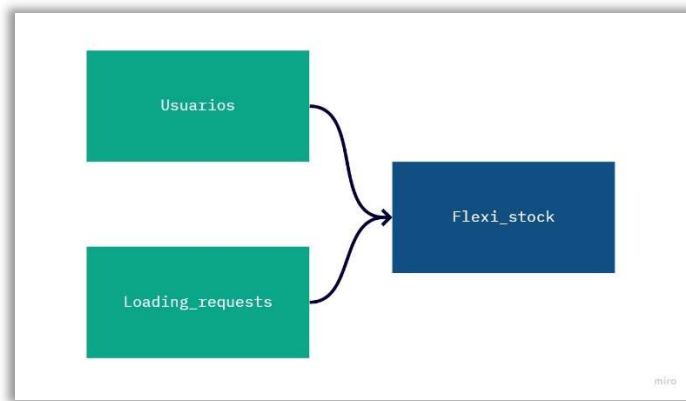
- 1 stand by - #F0E68C (Yellow) – **Status Condition of the Commercial Edit**
- 2 booking submitted or transferred - Blue (Green) oi #B0E0E6 - **Status Condition of the Commercial Edit**
- 3 status booking to be cancelled - #f5deb3 (Orange) - **Status Condition of the Commercial Edit**
- 4 booking Cancelled - #D8BFD8 (Purple)
- 5 fake - pre booking - #f5f5f5 (Light Gray) - **Pre-Status Condition**
- 6 real - #e8afaf (Red) - **Pre-Status Condition**

Status as Pre-Booking or Fake:

When the reservation presents this status, the icon  appears for the user. If the user needs to alter any information of the Booking without the validations of the logistic programming, simply keep the customer as CSA do Brasil and save the data desired to alter or include. No e-mail is sent in this case. For that, the user must perform the normal alteration process, following all relevant validations.

Place the list. Order by Routes, Carrier, Ports and ETA POL, respectively.

20. Flexi Stock



FlexiTank stock control table within the Tabex system.

Preconditions: Secondary Registration:

- ✓ Users
- ✓ Loading_requests

Subsidy to perform the following:

- ✓ none

20.1.1 Flexi Stock List – Current Stock

Consult all the Flexi Tanks available in the company's stock.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/flexistock/list
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	flexiStock.list
View Parameters:	compact(flexiStock)
Method:	FlexiStockController@index
Route Name:	listFlexiStockOut
Mandatory Folder:	/flexiStock

20.1.2 Flexi Stock List Output

Consult all the Flexi Tanks available in the company's stock.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/flexistock/list-out
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	flexiStock.listOut
View Parameters:	compact(flexiStock)
Method:	FlexiStockController@indexOut
Route Name:	listFlexiStockOut
Mandatory Folder:	/flexiStock

20.1.3 + New - Flexi Stock Input

Form containing fields with Flexitank input details and other parts.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/flexistock/create-in
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	flexiStock.createIn
Method:	FlexiStockController@createIn
Route Name:	createFlexiStockEntrada

20.1.3.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/flexistock/store
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	FlexiStockController@store
Route Name:	storeFlexiStock

20.1.4 + New - Flexi Stock Output

Form containing fields with Flexitank output details and other parts.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/flexistock/create-out
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	flexiStock.createOut
Method:	FlexiStockController@createOut
Route Name:	createFlexiStockSaida

20.1.4.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/flexistock/storeOut
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	FlexiStockController@storeOut
Route Name:	storeFlexiStockOut

20.1.5 Edit - Entrada

Modal is displayed with a description field to edit the selected record, in addition to the status button to activate or deactivate the record.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/flexistock/{id}/edit-in
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	

View:	flexiStock.editIn
Method:	FlexiStockController@editIn
Route Name:	editFlexiStockEntrada

20.1.5.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/flexistock/{id}/update
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	FlexiStockController@update
Route Name:	updateFlexiStock

20.1.6 Delete

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/flexistock/{id}
HTTP Method:	GET
Sending Parameters:	FlexiStock
Internal Information:	
View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	FlexiStockController@destroy
Route Name:	deleteFlexiStock

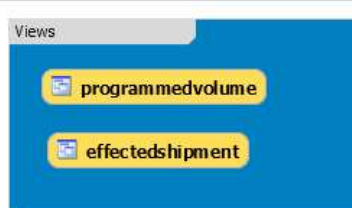
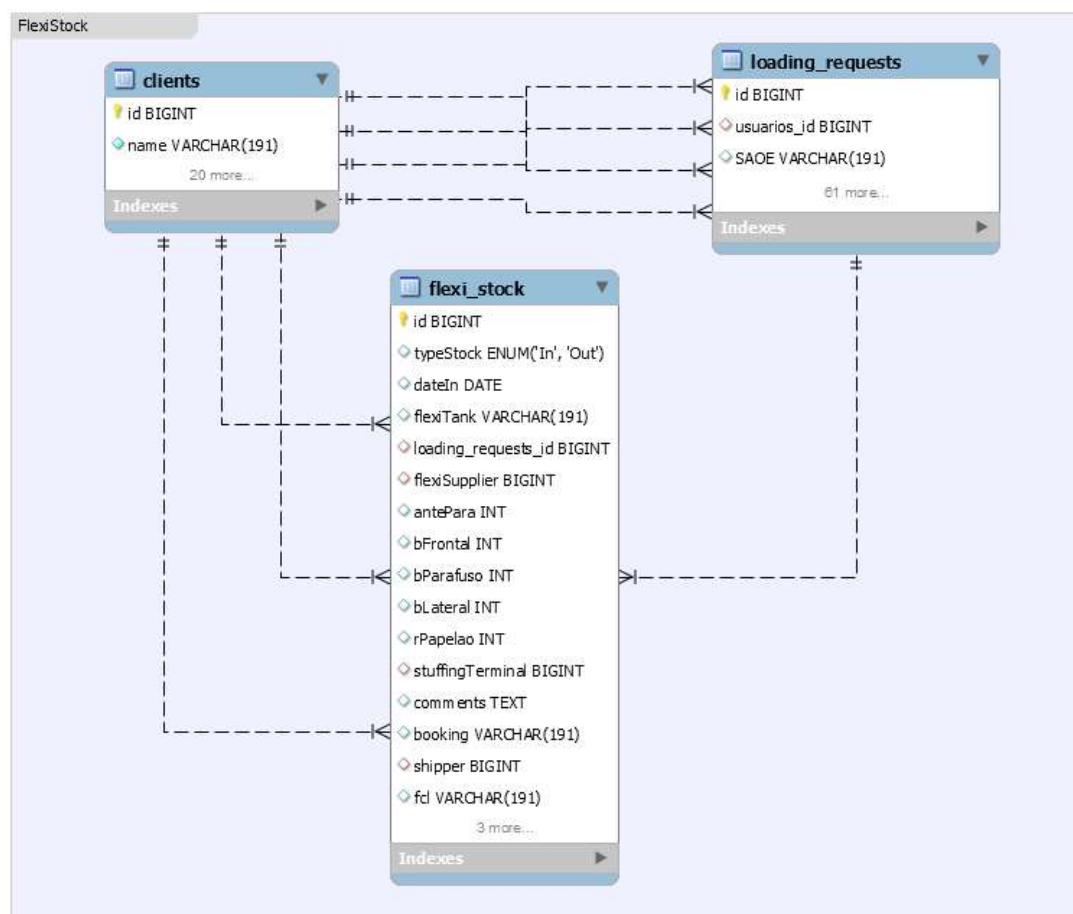
20.1.7 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:										
Table	flexi_stock									
Description	Registration of FlexiTank Stock									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
typeStock	enum('In', 'Out')	Yes	NULL	x	x	x				
dateIn	date	Yes	NULL	*	*	x				
flexiTank	varchar(191)	Yes	NULL	x	x	x				
loading_requests_id	bigint(20)	Yes	NULL	x	x	x		x		
flexiSupplier	bigint(20)	Yes	NULL	*	*	x				
antePara	int(11)	Yes	NULL			x				
bFrontal	int(11)	Yes	NULL			x				
bParafuso	int(11)	Yes	NULL			x				
bLateral	int(11)	Yes	NULL			x				
rPapelaio	int(11)	Yes	NULL			x				
stuffingTerminal	bigint(20)	Yes	NULL			x				

comments	text	Yes	NULL			X				
booking	varchar(191)	Yes	NULL	*	*	X				
shipper	bigint(20)	Yes	NULL			X				
fcl	varchar(191)	Yes	NULL	*	*	X				
montagemDate	date	Yes	NULL	*	*	X				
created_at	timestamp	Yes	NULL	X	X					
updated_at	timestamp	Yes	NULL		X					

EER Diagram:



Rules:

Field **dateIn** is mandatory in the input stock screen.

Field **FlexiTank** is mandatory and unique. There may only be one input and one output number.

Field **Booking** is mandatory for stock output. The user must indicate in which logistical schedule the Booking has been linked for control purposes.

Field **fcl** is only displayed on the output screen.

Field **montagemDate** is only displayed on the output screen, mandatory.

Field **typeStock** is represented by the values 'IN' or 'OUT'.

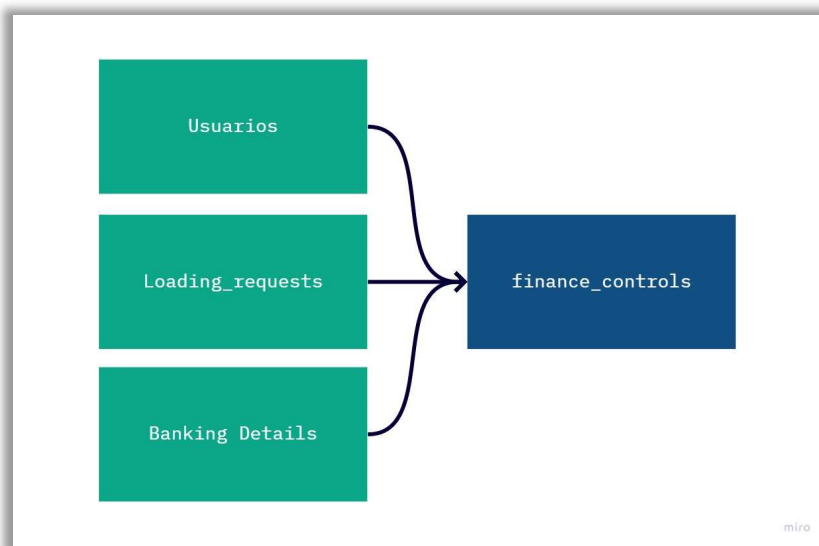
- IN = Stock Input.
- Out = Stock Output.

Should not contain the same number of FlexiTank 2 'IN' or 2 'OUT'. These are unique.

Report is sorted by subtotal by Stuffing Terminal and Supplier.

A general stock count is only displayed for visual monitoring purposes.

21. Finance



In order to meet this module requirement, Tables with the respective specifications were created.

FlexiTank stock control table within the Tabex system.

Preconditions: Secondary Registration:

- ✓ Users
- ✓ Loading_requests
- ✓ Banking Details

Subsidy to perform the following:

- ✓ Payment Request
- ✓ Finance Controls

21.1.1 Invoicing Pendencies – List

Consult all shipments not invoiced.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/finance/list
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	finance.list
View Parameters:	compact('registers','loading_empties','loading_stuffings', 'pag' , 'select','total')
Method:	FinanceController@index
Route Name:	listFinance
Mandatory Folder:	/finance

21.1.2 Invoiced Processes – List

Consult all shipments not invoiced.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/finance/InvoicedProcesses
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	

View:	finance.invoicedProcesses
View Parameters:	compact('registers','loading_empties','loading_stuffings', 'pag' , 'select','total','counts')
Method:	FinanceController@InvoicedProcesses
Route Name:	InvoicedProcesses
Mandatory Folder:	/finance

21.1.3 Register Clearance

Screen to clear shipments not invoiced, Registration made with the options N/A, Done and Pending.

21.1.3.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/invoice/store
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	finance.invoice
Method:	FinanceController@store
Route Name:	storeInvoice

21.1.4 Edit Clearance

Screen to edit shipments not invoiced, Registration made with the options N/A, Done and Pending.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/invoice/{id}/edit
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	finance.invoice
Method:	FinanceController@edit
Route Name:	editInvoice

21.1.4.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/invoice/{id}/update
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	finance.invoice
Method:	FinanceController@update
Route Name:	updateInvoice

21.1.5 Delete

A modal screen is displayed to confirm the cancellation of the record.

The delete action updates the status of the record to 'Inactive'. Deleting the record is not possible because there is a relationship between tables in the system and also to keep the records' history.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/invoice/{id}
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	

View:	Modal -> include/ modal_deletar.blade.php
View Parameters:	data-catid = catid
Method:	FinanceController@destroy
Route Name:	deleteInvoice

21.1.6 UML

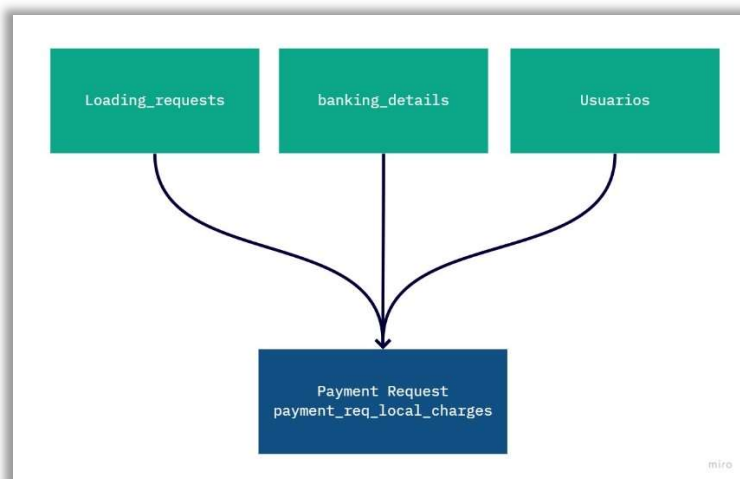
In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:										
Table	finance_controls									
Description	Registration of the Invoicing Control									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
loading_requests_id	bigint(20)	Yes	NULL	x	x		x			
SAOE	varchar(191)	Yes	NULL	x	x					
brazilianInvoice	enum('Done', 'Pending', 'N/A')	Yes	NULL	*	*	x				
brazilianInvoiceDate	date	Yes	NULL	*	*	x				
comments	text	Yes	NULL			x				
suisselInvoice	enum('Done', 'Pending', 'N/A')	Yes	NULL	*	*	x				
suisselInvoiceDate	date	Yes	NULL	*	*	x				
commentsSuisse	text	Yes	NULL			x				
status	varchar(20)	Yes	Pending							
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

Rules:

The Fields **brazilianInvoice** and **brazilianInvoiceDate** refers to Invoicing performed in Brazil.
The Fields **suisselInvoice** and **suisselInvoiceDate** refers to Invoicing performed Outside of Brazil.
Status Type: 'Done', 'Pending', 'N/A'
In case both fields are filled, **brazilianInvoice** and **suisselInvoice**, the registration is displayed on the Invoiced Processes screen.

22. Payment Request - Routine



Routine for payment requests of local fees.

Preconditions: Secondary Registration:

- ✓ Users
- ✓ Loading_requests
- ✓ Banking Details
- ✓ Origin Freight Rates

Subsidy to perform the following:

- ✓ Payment Request
- ✓ Finance Controls

22.1.1 Payment request Dashboard

Only the Financial Area has access to the list of payment requests. After the operational team performs the request process, a record is created to monitor the process, until a payment receipt is attached for the status to alter from 'Requested' to 'Done'.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	paymentrequest/list
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	loading.paymentRequest.list
View Parameters:	compact('registers','select','count_column','returnvalues')
Method:	PaymentRequestOperationalController@index
Route Name:	listPaymentRequest
Mandatory Folder:	/ loading/paymentRequest

22.1.2 Shipped List – Payment Request - Create

The payment request is performed through the Shipment Screen, using the icon:



Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/paymentrequest/{id}/create
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	

View:	loading.paymentRequest.paymentRequest
View Parameters:	compact('register','loadingRequest','select','attachments','item','bank')
Method:	PaymentRequestOperationalController@create
Route Name:	createPaymentRequest
Mandatory Folder:	/loading/paymentRequest

22.1.2.1 Store Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/paymentrequest/store
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
Method:	PaymentRequestOperationalController@store
Route Name:	storePaymentRequest

22.1.3 Edit

Screen to edit and/or attach the payment receipt to the process.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/paymentrequest/{id}/edit
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	loading.paymentRequest.edit
Method:	PaymentRequestOperationalController@edit
Route Name:	editPaymentRequest

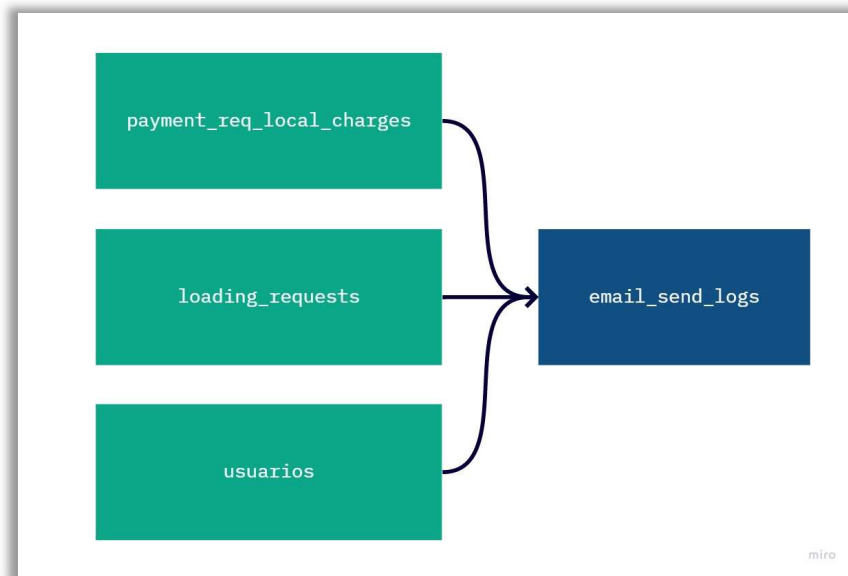
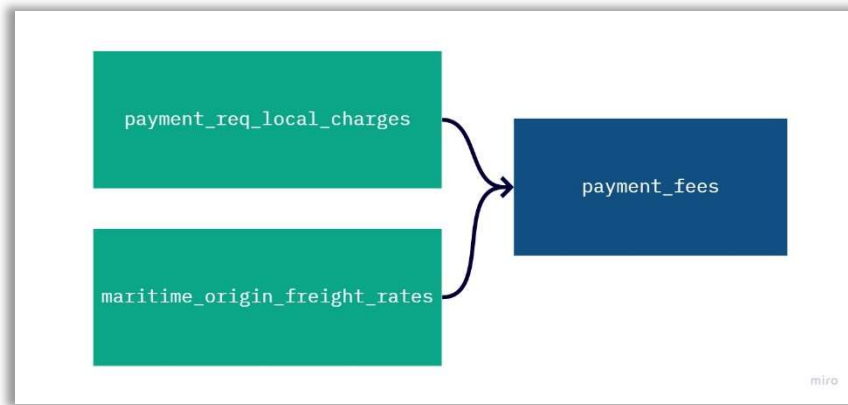
22.1.3.1 Update Info

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/paymentrequest/{id}/update
HTTP Method:	['get', 'post']
Sending Parameters:	id
Internal Information:	
Method:	PaymentRequestOperationalController@update
Route Name:	updatePaymentRequest

22.1.4 Delete

none

22.1.5 UML



In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:

Table	payment_req_local_charges									
Description	Registration of Payment Requests									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
loading_requests_id	bigint(20)	Yes	NULL	x	x			x		
SAOE	varchar(191)	No		x	x					
banking_details_id	bigint(20)	Yes	NULL	x	x			x		
banking_details_id2	bigint(20)	Yes	NULL		x			x		
users_id	bigint(20)	Yes	NULL	x	x			x		
dollarRate	decimal(10,2)	Yes	NULL							
amount	decimal(10,2)	Yes	NULL	x	x					
containerQty	int(11)	Yes	NULL	x	x					
blQty	int(11)	Yes	NULL	x	x					
CorretionFee	varchar(10)	Yes	No							

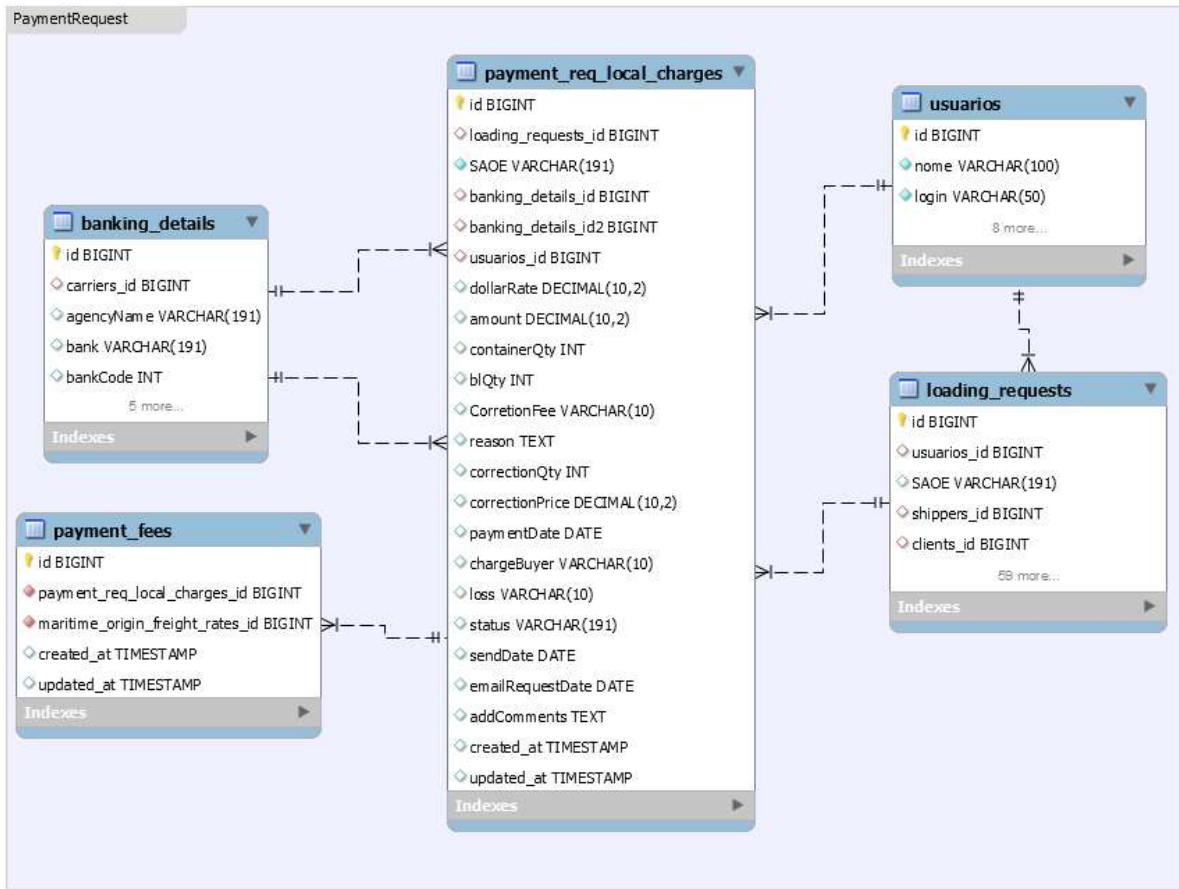
reason	text	Yes	NULL							
correctionQty	int(11)	Yes	NULL							
correctionPrice	decimal(10,2)	Yes	NULL							
paymentDate	date	Yes	NULL							
chargeBuyer	varchar(10)	Yes	No							
loss	varchar(10)	Yes	No							
status	varchar(191)	Yes	NULL	x	x					
sendDate	date	Yes	NULL	x						
emailRequestDate	date	Yes	NULL							
addComments	text	Yes	NULL							
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL		x					

Table	payment_req_local_charges									
Description	Registration of Payment Requests									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
payment_req_local_charges_id	bigint(20)	No		x				x		
maritime_origin_freight_rates_id	bigint(20)	No		x				x		
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL							

Table	email_send_logs									
Description	Registration of e-mail submission records									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
usuario_id	varchar(191)	Yes	NULL	x				x		
userLogin	varchar(191)	Yes	NULL	x						
SAOE	varchar(191)	Yes	NULL							
loading_requests_id	varchar(191)	Yes	NULL	x				x		
payment_req_local_charges_id	varchar(191)	Yes	NULL					x		
emailRoute	varchar(191)	Yes	NULL							
addcomments	text	Yes	NULL							
addInfo	varchar(500)	Yes	NULL							
email	varchar(500)	Yes	NULL	x						
cc	varchar(500)	Yes	NULL	x						
bcc	varchar(500)	Yes	NULL	x						
userMail	varchar(500)	Yes	NULL	x						
created_at	timestamp	Yes	NULL	x						

updated_at	timestamp	Yes	NULL						
------------	-----------	-----	------	--	--	--	--	--	--

EER Diagram:



Tabelas Auxiliares - Logs

- email_send_logs
- loading_links

Views

- programmedvolume
- effecteds shipment

Rules:

Fees are presented according to the respective records.

Only Origin Type fees are displayed on this screen.

User is free to add the correction rate, quantity and price.

The fee is calculated according to the registration, either by container or by quantity of BL.

Dollar rate, is a field to indicate the dollar rate. If the rate is in dollar 'USD,' the system will automatically convert, according to the specified rate, the foreign currency to Real.

In order to proceed with the payment request, a previous registration of the bank accounts of the carriers is required.

Only 2 bank accounts can be informed per payment request.

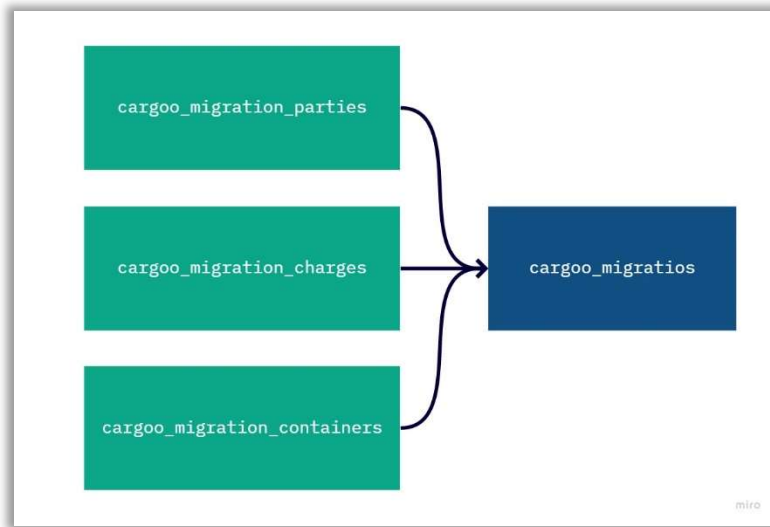
Rate values are not altered on this screen. There must be a prior registration for this.

An e-mail is sent to the person responsible for the payment and the following information is saved in a log table: date, time, id and user who sent the e-mail, as shown in the table email_send_logs

There is a Table for linking (N:N) between Payment Request and Origin Freight and shipowners. The registration is performed following the database integrity sequence.

23. Tariff Costs

24. Webservice Routine



Data migration screen between Cargoo x Tabex of the booking requests displayed for internal control.

24.1.1 Migration Data – List

In order to perform a quick consultation in the system of the records to be migrated or already migrated through the integration.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/migration/list
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	Services.listMigration
View Parameters:	compact('registers')
Method:	ServicesController@index
Route Name:	listMigration
Mandatory Folder:	/ services

24.1.2 Edit

Editing is performed only in the registry status in order for the administrator to resend the migration to the Tabex system.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/migration/{id}/edit
HTTP Method:	GET
Sending Parameters:	id
Internal Information:	
View:	services.editMigration
View Parameters:	compact('registers')
Method:	ServicesController@edit
Route Name:	editMigration
Mandatory Folder:	/ services

24.1.2.1 Update

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/migration/{id}/update
HTTP Method:	['get', 'post']
Sending Parameters:	id
Internal Information:	
Method:	ServicesController@update
Route Name:	updateMigration

24.1.3 Migration Request Manual Update

Through the application, we can manually initiate the update of the micro data migration service between migrated data through the integration of Cargoo x Tabex data.

Item	Value
Server:	http://sistema.tabex.com.br
Endpoint:	/microservice/list
HTTP Method:	GET
Sending Parameters:	null
Internal Information:	
View:	services.consumingMicroservice
View Parameters:	compact('registers')
Method:	RequestMigrationData@viewConsumingService
Route Name:	viewConsumingService
Mandatory Folder:	/ services

24.1.4 UML

In order to meet this module requirement, Tables with the respective specifications were created:

Data Dictionary:

Table	cargoo_migrations									
Description	Registry of data of Migration Cargoo x Tabex									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No					x			x
bookingNumber	varchar(191)	Yes	NULL							
cargooReference	varchar(191)	Yes	NULL	x						
bINumber	varchar(191)	Yes	NULL							
actualTeus	varchar(191)	Yes	NULL							
plannedTeus	varchar(191)	Yes	NULL							
incoterm	varchar(191)	Yes	NULL							
movmentType	varchar(191)	Yes	NULL							
commodityCode	varchar(191)	Yes	NULL							
commodityName	varchar(191)	Yes	NULL							
commodityDescription	varchar(191)	Yes	NULL							
commodityArticle	varchar(191)	Yes	NULL							
packages	varchar(191)	Yes	NULL							

phones	varchar(191)	Yes	NULL							
type	varchar(191)	Yes	NULL							
remarks	varchar(191)	Yes	NULL							
created_at	timestamp	Yes	NULL							
updated_at	timestamp	Yes	NULL							

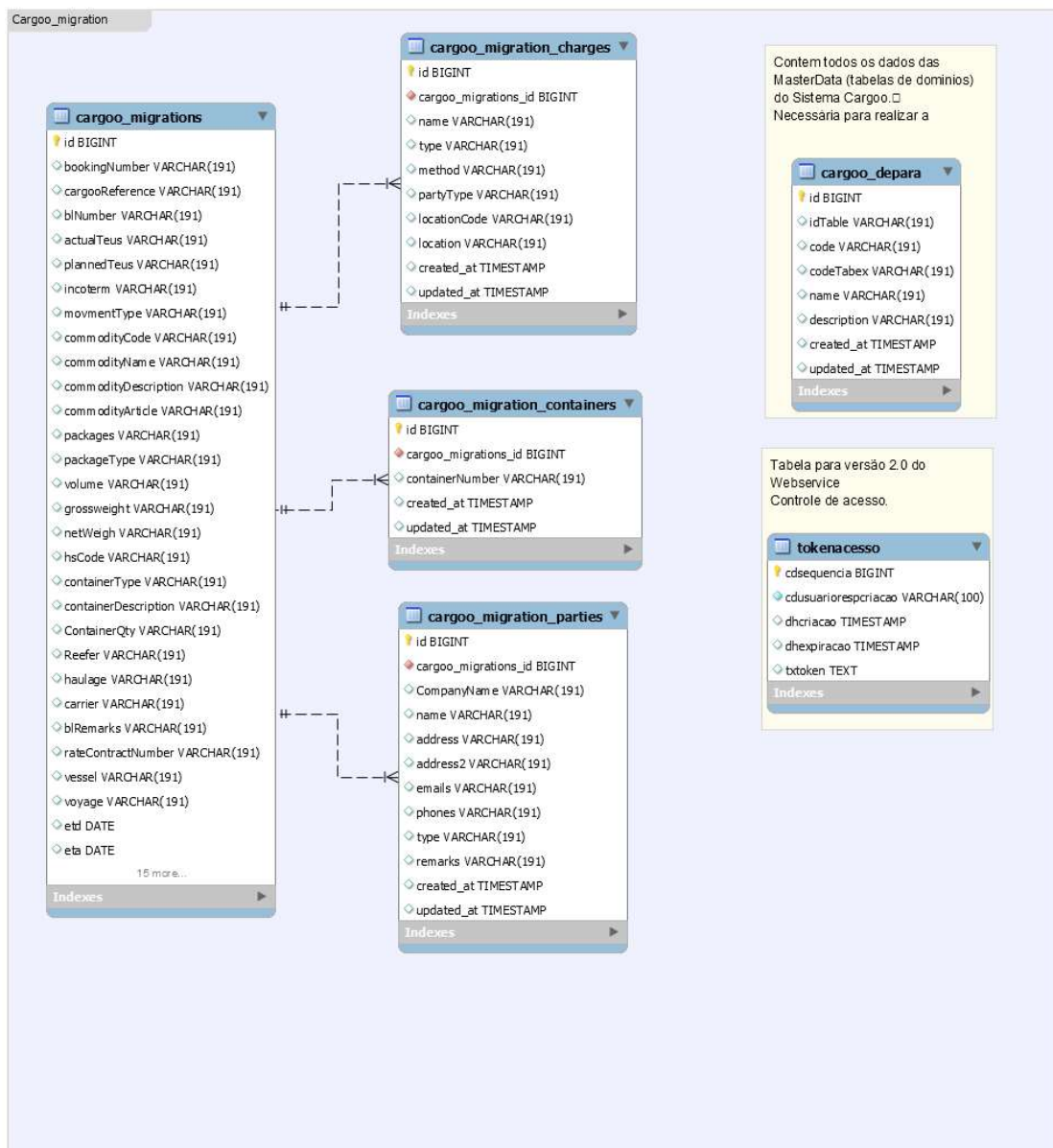
Table	cargoo_migration_charges									
Description	Registry of data of Migration Cargoo x Cargoo Charges									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
cargoo_migrations_id	bigint(20)	No		x						
name	varchar(191)	Yes	NULL							
type	varchar(191)	Yes	NULL							
method	varchar(191)	Yes	NULL							
partyType	varchar(191)	Yes	NULL							
locationCode	varchar(191)	Yes	NULL							
location	varchar(191)	Yes	NULL							
created_at	timestamp	Yes	NULL							
updated_at	timestamp	Yes	NULL							

Table	cargoo_migration_containers									
Description	Registry of data of Migration Cargoo x Cargoo Containers									
Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
cargoo_migrations_id	bigint(20)	No		x			x			
name	varchar(191)	Yes	NULL							
type	varchar(191)	Yes	NULL							
method	varchar(191)	Yes	NULL							
partyType	varchar(191)	Yes	NULL							
locationCode	varchar(191)	Yes	NULL							
location	varchar(191)	Yes	NULL							
created_at	timestamp	Yes	NULL	x						
updated_at	timestamp	Yes	NULL							

Table	cargoo_depara									
Description	Table of From/To of the migration codes									

Field	Type	Null	Value Allow	Mandatory		Filed (layout)	PK	FK	NN	AI
				Insert	Edit					
id	bigint(20)	No		x			x			x
idTable	varchar(191)	Yes	NULL							
code	varchar(191)	Yes	NULL							
codeTabex	varchar(191)	Yes	NULL							
name	varchar(191)	Yes	NULL							
description	varchar(191)	Yes	NULL							
created_at	timestamp	Yes	NULL							
updated_at	timestamp	Yes	NULL							

EER Diagram:



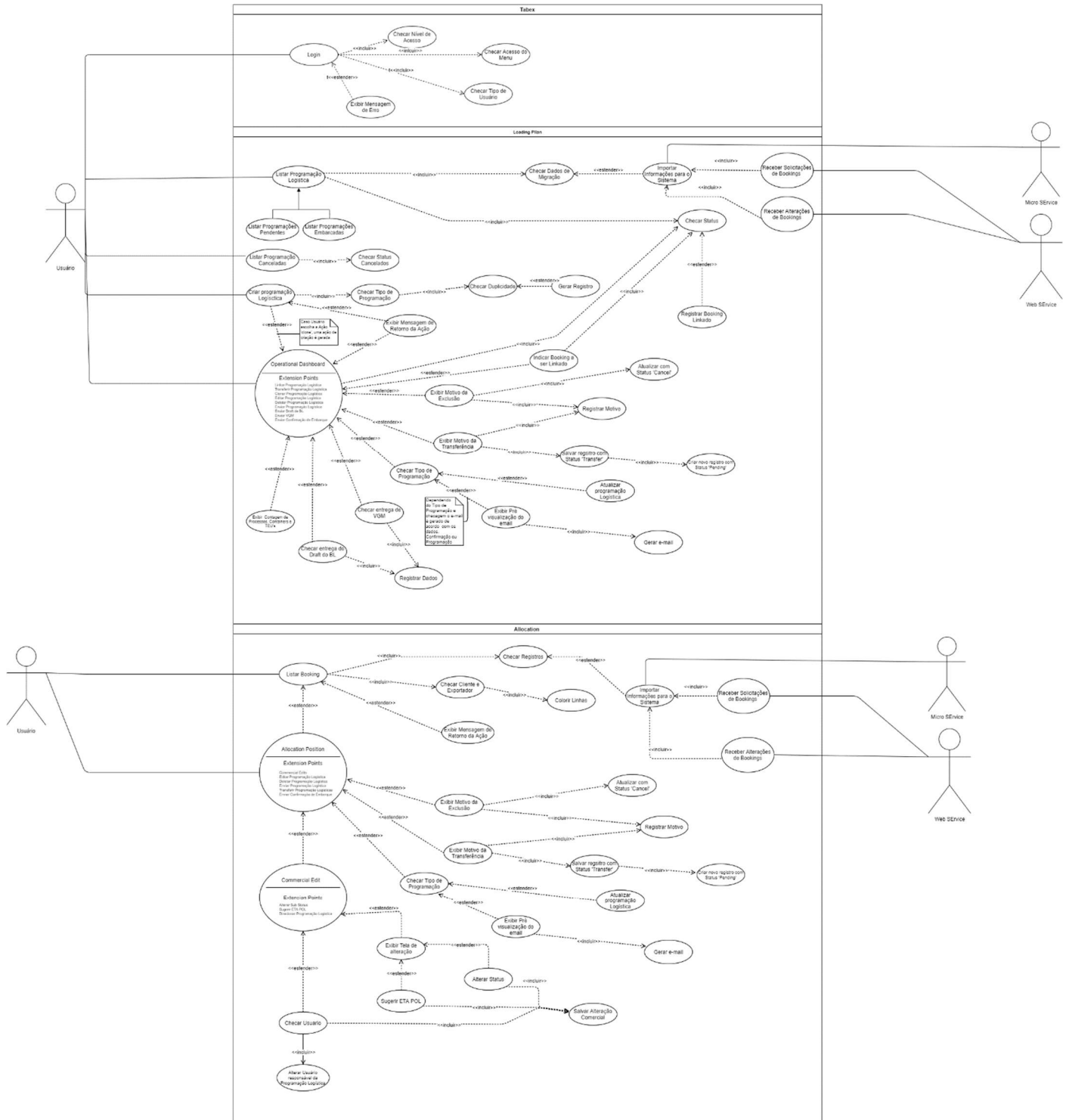
Rules:

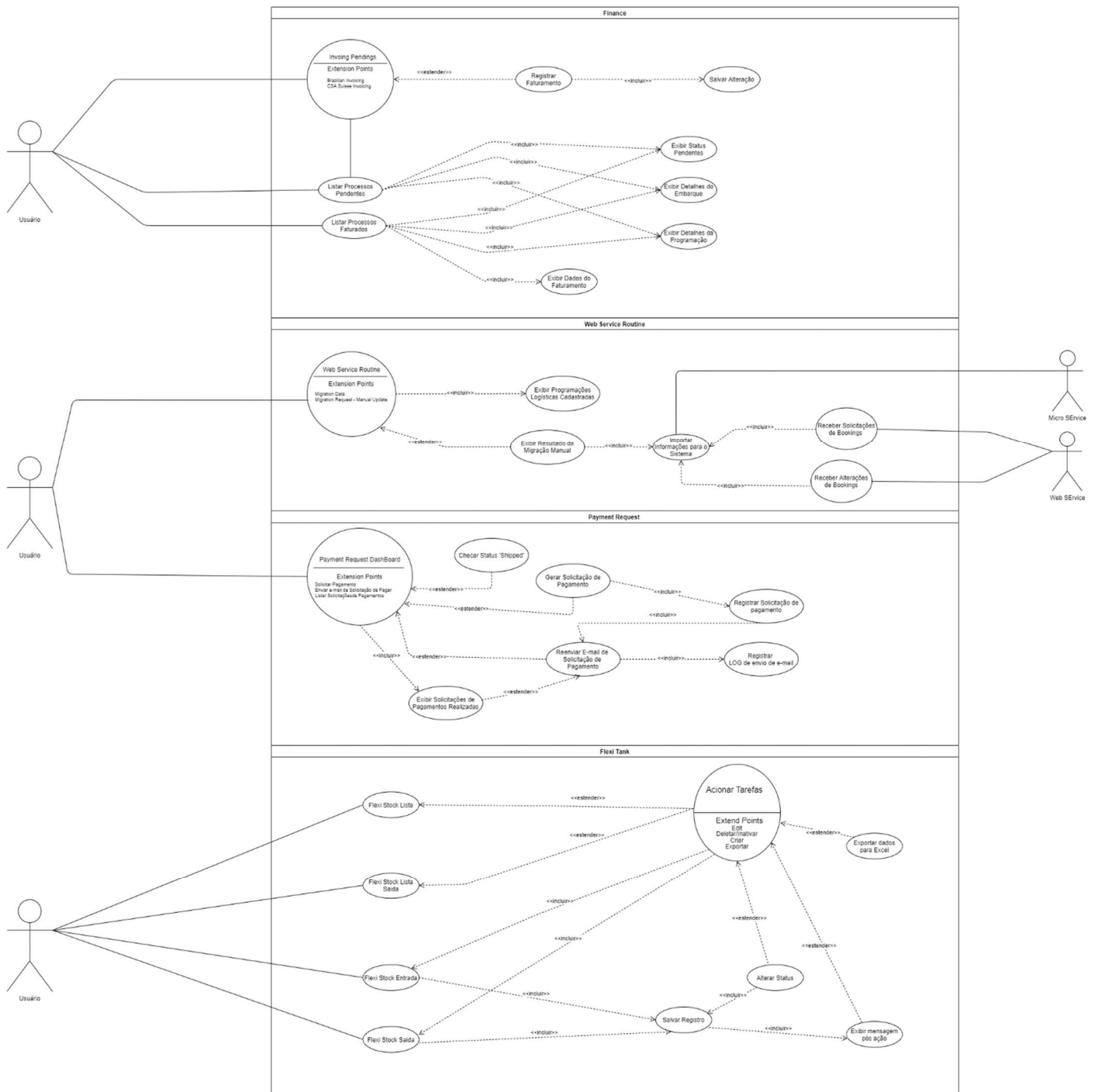
Every time a record is migrated, a copy is created for the production bank.

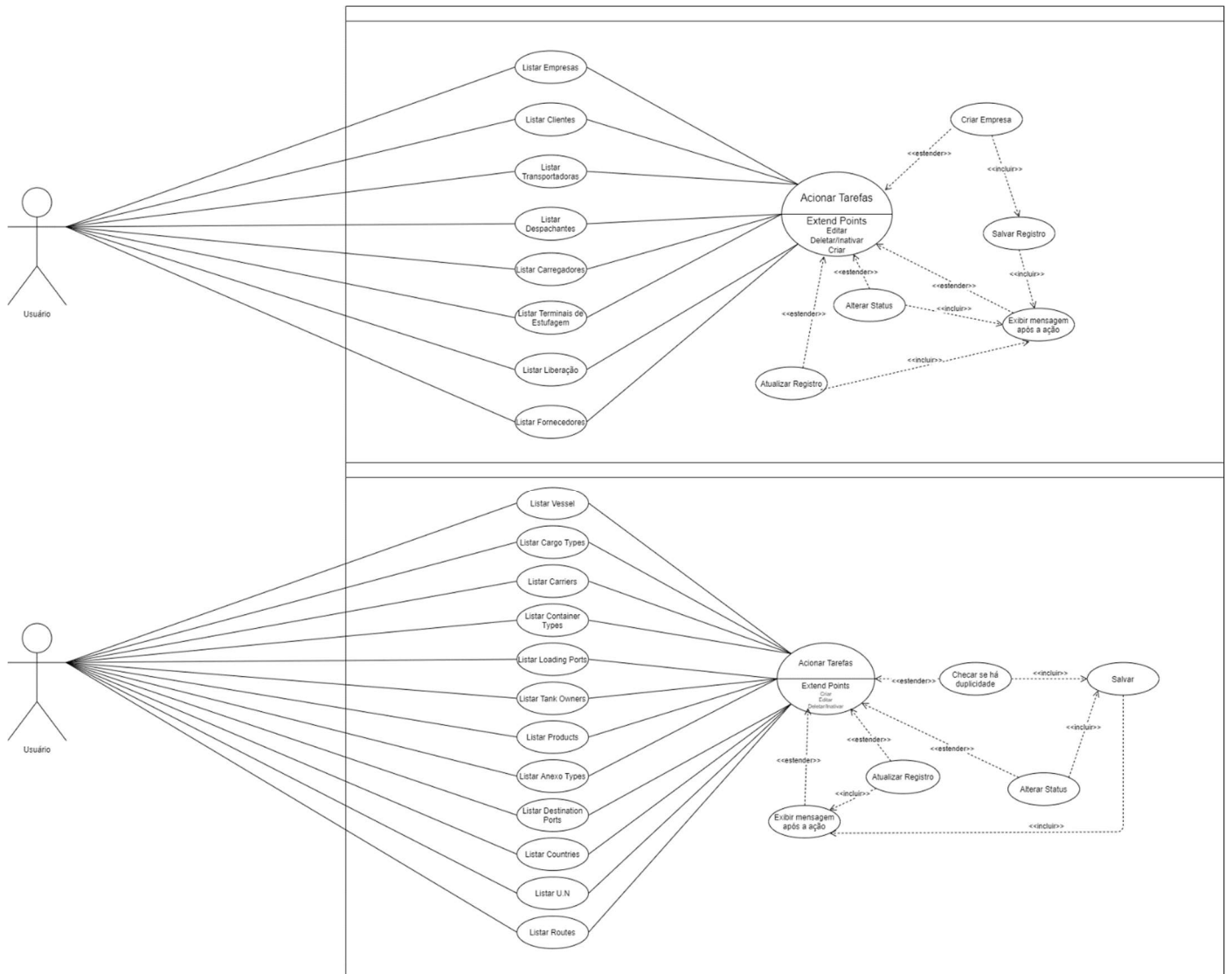
For data migration, the Mandatory Field is the Cargo Reference.

24.1.5 General Information –

24.1.6 Use Case - Tabex System







GLOSSARY

A

Anexo	Attachment
Armazenar Informações	Store Info
Atualizar Draft BL + VGM	Update Draft BL + VGM
Atualizar Informações	Update Info

B

Bill of Lading	BL – Bill of Lading
----------------	---------------------

C

Cancelar	Cancel
Cliente	Client
Clonar; Duplicar	Clone
Controlador de Envio de E-mail	E-mail Send Controller
Criar	Create

D

Data de Envio	Sent Date
Detalhes da Solicitação de Embarque	Booking Details
Draft de BL Enviado	Draft BL Sent

E

Editar	Edit
Embarcação	Vessel
Enviar E-mail	Send an E-mail

H

Horário de Envio	Time Sent
------------------	-----------

I

Informação Interna	Internal Information
--------------------	----------------------

L

Lista de Embarques	Loading List
Lista de Pendências	Pending List
Lista Embarcados	Shipped List
Lista Itens Cancelados	Cancelled List

M

Método	Method
Método HTTP	HTTP Method

N

Não	No
Nome do Usuário	User Name
Nome Fantasia	Trade Name; Fantasy Name
Nome Roteador	Route Name
Nulo	Null
Número da Solicitação de Embarque	Booking Number

P

Parâmetros de Envio	Sending Parameters
Período de Data	Date Period
Plano Logístico	Logistics Plan
Ponto de Extremidade	Endpoint
Pré-Draft	Pre-Draft
Pré-visualização do E-mail	Email Preview

S

Servidor	Server
Sim	Yes
Solicitações de Embarque	Booking Plan

T

Transferência para Outra Solicitação de Embarque	Transfer to Another Booking
--	-----------------------------

U

Usuário	User
---------	------

V

Valor	Value
Vincular	Link
Vincular a Outra Solicitação de Embarque	Link to Another Booking
Visualizar Parâmetros	View Parameters



**Thank you for using the CSA
Tabex System!**

CSA tabex system

Field Rules Control