



Universidade do Porto

Faculdade de Engenharia

FEUP

SISTEMAS DE INFORMAÇÃO

4º ANO DO MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA E
COMPUTAÇÃO

Sales Force Automation

Relatório de Implementação

Estudantes:

Turma 4MIEIC02, Grupo C

Filipa RAMOS

- up201305378@fe.up.pt

Gil DOMINGUES

- up201304646@fe.up.pt

Pedro PONTES

- up201305367@fe.up.pt

Pedro MELO

- up201305618@fe.up.pt

15 de Dezembro de 2016

Conteúdo

1	Introdução	2
2	Especificação	4
2.1	Descrição	4
2.2	Core Views	4
2.2.1	Client Page: Display of client information and transaction history	4
2.2.2	Homepage: Overview of schedule	5
2.2.3	Meeting Page: Overview of meeting	6
2.2.4	New Client Page: Create page for new potencial client	7
2.2.5	New Meeting Page: Schedule meeting with client	8
2.2.6	New Order Page: Sell products to client	9
2.2.7	Order Status Page: Overview of order status	10
2.2.8	Product Page: Overview of product	11
2.2.9	Catalog Page: List of products	12
2.2.10	Administration Page: Overview of sales and sales team	13
2.3	Funcionalidades	14
3	Implementação	15
3.1	Interoperabilidade	15
3.1.1	Clientes	15
3.1.2	Reuniões	16
3.1.3	Armazéns	17
3.1.4	Transações	18
3.1.5	Artigos	19
3.1.6	Administrador	20
3.2	Caminhos	21
4	Arquitetura	21
5	Especificação vs Implementação	21
6	Conclusões	21

1 Introdução

Muito tem sido dito acerca do papel da tecnologia no *marketing* e como esta pode substituir os processos de venda tradicionais. No entanto, os representantes de vendas ainda são fundamentais no fecho de negócios: falando com clientes por telefone ou pessoalmente, são eles quem «vende» aos clientes os méritos e benefícios de um dado produto, levando à geração de lucros. A tecnologia não consegue substituir, por completo, a interação humana.

Com efeito, a equipa de vendas continua a ser a face de uma empresa – ainda que nuns setores mais que noutros. Porque essas empresas despendem uma considerável quantidade de tempo e dinheiro com as equipas de vendas, importa geri-las de forma tão eficiente e eficaz quanto possível.

Para tal, existem soluções *SFA* – *Sales Force Automation* –, software que permite automatizar tarefas como controlo de inventário, processo de vendas, e registo de interações com clientes. São normalmente integradas na componente de *Customer Relationship Management (CRM)* de um sistema de *Enterprise Resource Planning (ERP)*, existindo várias alternativas no mercado: *SAP CRM (Sales)*, *Microsoft Dynamics CRM* e *Oracle Siebel* são alguns exemplos.

Realizado no âmbito da unidade curricular de Sistemas de Informação, o presente documento detalha a especificação de uma aplicação web que oferecerá as funcionalidades de uma solução *SFA*. Ao longo

do documento, é feita uma descrição do projeto, enumerando-se as funcionalidades a implementar, e identificam-se os pontos de ligação com o *ERP* Primavera.

2 Especificação

2.1 Descrição

//////////////////////////////////// O projeto consiste no desenvolvimento de uma aplicação web que oferecerá as funcionalidades de uma solução *SFA*. Existirá uma camada de autenticação, através da qual um supervisor e os representantes de vendas deverão autenticar-se, para que possam organizar a sua agenda, manter o registo dos seus contactos com clientes e registar ordens de venda.

A aplicação web será desenvolvida utilizando Node.js. Porque com este trabalho se pretende, também, adquirir conhecimentos no que respeita ao uso de um ERP – no caso, o Primavera – prevê-se, igualmente, a criação de um servidor back-end em *C#*, que fara a comuniação com o Primavera, passando a informação à aplicação front-end através de uma API RESTful.

2.2 Core Views

2.2.1 Client Page: Display of client information and transaction history

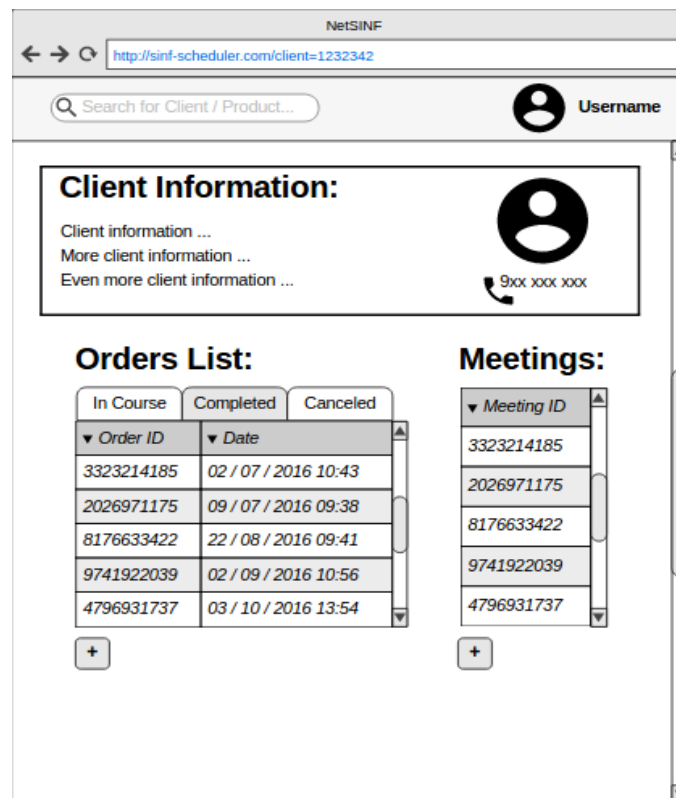


Figura 1: Core View da página de cliente.

Tabela 1: Especificação da Core View Client Page.

ID CORE VIEW	CLI_PAGE
INWARD PATHS	<ul style="list-style-type: none"> -Homepage; -Meeting Page; -Order Page.
OUTWARD PATHS	<ul style="list-style-type: none"> -Allows the user to see detailed client information; -Displays complete history of pending, completed and canceled orders; -Allows starting a new order with the client; - Display history of meetings with client; -Allows scheduling a new meeting with the client.
ELEMENTS OF THE CORE	Client Information Summary, Order Information Table, Context Tabs and New Order Button

2.2.2 Homepage: Overview of schedule

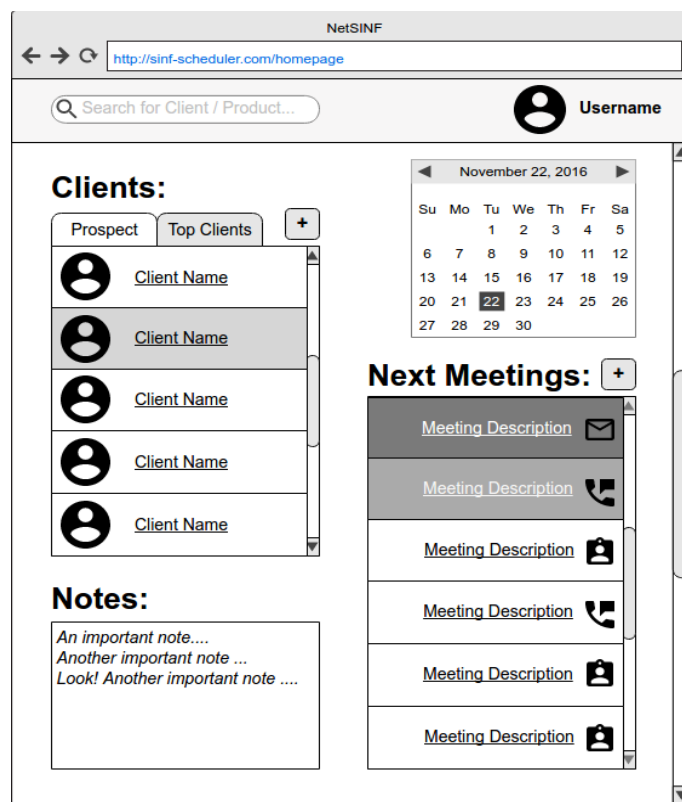


Figura 2: Core View da página inicial.

Tabela 2: Especificação da Core View Homepage.

ID CORE VIEW	HOM_PAGE
INWARD PATHS	-Sign in; -Home button
OUTWARD PATHS	-Allow user to quickly verify future meetings and take relevant notes if necessary; -Enables scheduling of new meetings with existing clients; -Allows user to add new clients and easily access their information.
ELEMENTS OF THE CORE	Clients Grid, Meetings list, Meetings calendar, Note text area

2.2.3 Meeting Page: Overview of meeting

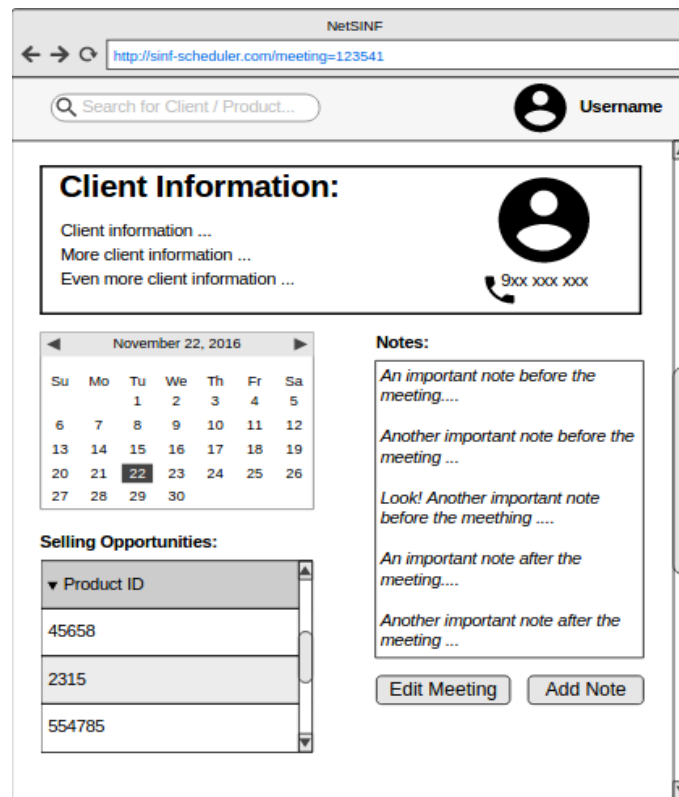


Figura 3: Core View da página das reuniões agendadas.

Tabela 3: Especificação da Core View Meeting Page.

ID CORE VIEW	VIE_PAGE
INWARD PATHS	-Homepage; -Client Page.
OUTWARD PATHS	-Allows the user to see detailed client information; -Allows the user to take relevant notes after the meeting; -Shows the detailed list of selling opportunities proposed to the client.
ELEMENTS OF THE CORE	Client Information Summary, Meeting Calendar, Notes Text Area and Selling Opportunities List

2.2.4 New Client Page: Create page for new potencial client

Figura 4: Core View da página para adicionar um cliente.

Tabela 4: Especificação da Core View New Client.

ID CORE VIEW	NEW_CLI
INWARD PATHS	-Homepage.
OUTWARD PATHS	-Allows the creation of new clients.
ELEMENTS OF THE CORE	New Client Form

2.2.5 New Meeting Page: Schedule meeting with client

The screenshot shows a web browser window with the URL `http://sinf-scheduler.com/new_meeting`. The page title is "Create New Meeting:". The form contains the following sections:

- Description:** A text area with placeholder text "Metting details to be shown in Homepage".
- Date / Time:** A date/time picker showing "4/22/2012 10:45".
- Client:** A list of client entries, each with a profile icon and the text "Client Name".
- Selling Opportunities:** A section with a "+" button and a list of product IDs: 45658, 2315, and 554785.
- Notes:** A text area with placeholder text: "An important note before the meeting....", "Another important note before the meeting ...", and "Look! Another important note before the meething".

Figura 5: Core View da página para adicionar uma reunião.

Tabela 5: Especificação da Core View New Meeting Page.

ID CORE VIEW	NEW_MEE
INWARD PATHS	-Homepage; -Client Page.
OUTWARD PATHS	-Enables the scheduling of new meetings with clients; -Verifies if different meetings do not overlap each other; -Enables to select the products to sell and access their respective pages; -Enables to take important notes before the meeting.
ELEMENTS OF THE CORE	Description Text Area, Meeting Data Picker, Client List and Selling Opportunities List

2.2.6 New Order Page: Sell products to client

NetSINF

http://sinf-scheduler.com/new_transaction

Search for Client / Product...

Username

Client Information:

Client information ...
More client information ...
Even more client information ...

9xx xxx xxx

Order Information:

Product ID	Quantity	Price
214185	02	\$4.00
1175	09	\$2.50
TOTAL:		\$6.50

Add Product
Edit Product
Remove Product

Issue Order

Figura 6: Core View da página para adicionar um pedido de um cliente.

Tabela 6: Especificação da Core View New Order Page.

ID CORE VIEW	NEW_TRA
INWARD PATHS	-Client Page; -Meeting Page.
OUTWARD PATHS	-Allows user to see specific information about the client; -Enables the creation of complete orders.
ELEMENTS OF THE CORE	Client Information Summary, New Order Table and Action Buttons

2.2.7 Order Status Page: Overview of order status

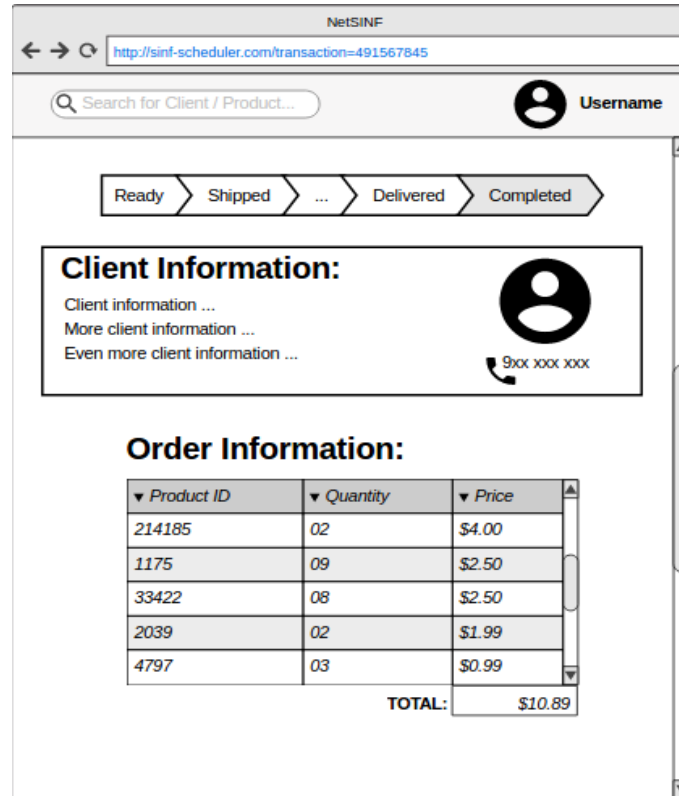


Figura 7: Core View da página onde se visualiza um pedido de um cliente.

Tabela 7: Especificação da Core View Order Status Page.

ID CORE VIEW	TRA_PAGE
INWARD PATHS	-Client Page.
OUTWARD PATHS	-Allows user to see the order status; -Enables user to access client information; -Enables user to see detailed information of order.
ELEMENTS OF THE CORE	Status Bar, Client Information Summary and Order Information Table

2.2.8 Product Page: Overview of product

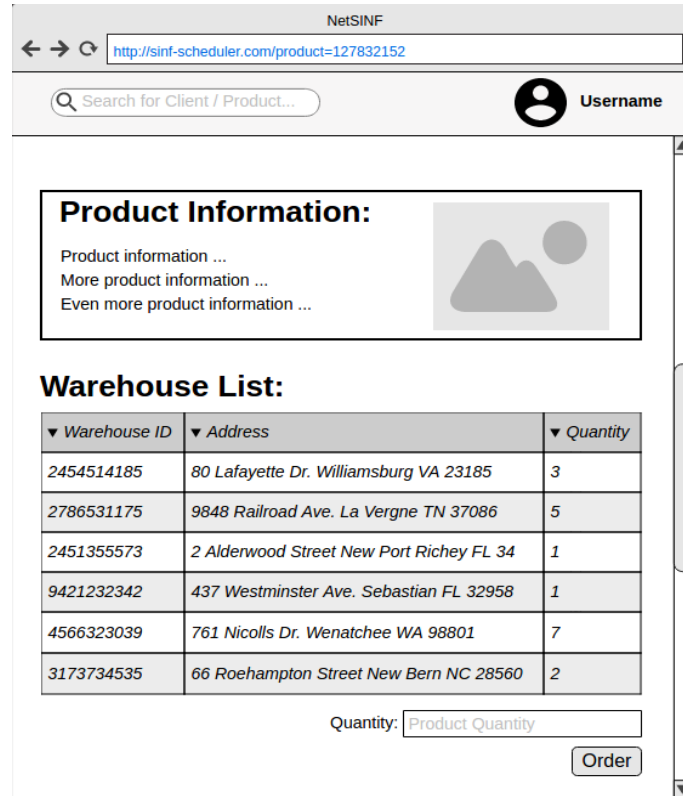


Figura 8: Core View da página de um produto e a sua disponibilidade em cada armazém.

Tabela 8: Especificação da Core View Product Page.

ID CORE VIEW	PRO_PAGE
INWARD PATHS	-Order Page; -Meeting Page.
OUTWARD PATHS	-Enables the user to see detailed information about a product; -Enables the user to check availability of a product in several warehouses and respective stock; -Enables user to request a defined quantity of product units.
ELEMENTS OF THE CORE	Product Information Summary and Warehouse Table

2.2.9 Catalog Page: List of products

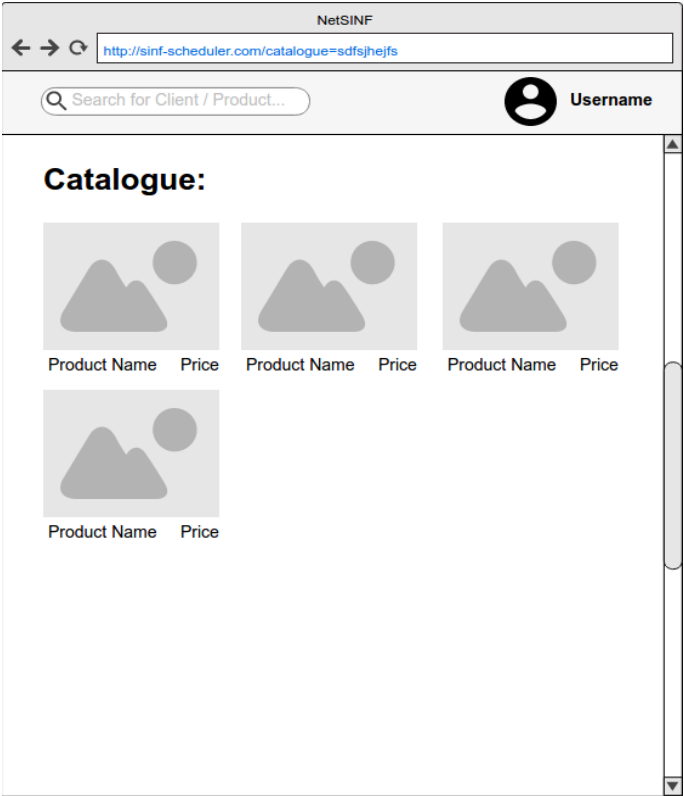


Figura 9: Core View da página do catálogo.

Tabela 9: Especificação da Core View Catalog Page.

ID CORE VIEW	CAT_PAGE
INWARD PATHS	-Meeting Page; -Administration Page; -Search.
OUTWARD PATHS	-Allows the user to see the products for sale in detail.
ELEMENTS OF THE CORE	Product List

2.2.10 Administration Page: Overview of sales and sales team

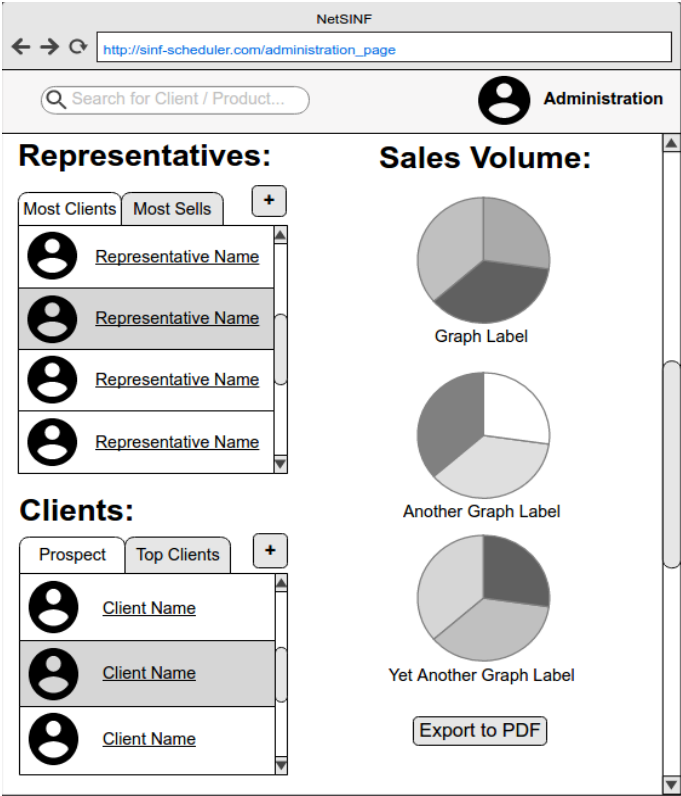


Figura 10: Core View da página de administração.

Tabela 10: Especificação da Core View Administration Page.

ID CORE VIEW	ADMIN_PAGE
INWARD PATHS	-Sign In.
OUTWARD PATHS	-Allows the user to see the sales representatives with the most clients or with the largest sales volume; -Allows the user to see prospect clients or with the largest sales volume; -Enables the user to see an overview of sales, including the best selling products.
ELEMENTS OF THE CORE	Sales Representative List, Clients List and Sales Overview

2.3 Funcionalidades

Tabela 11: Descrição das funcionalidades do software.

Funcionalidade	Finalidade	CoreView
REGISTER_REPRESENTATIVE	Registrar um representante de vendas	ADMIN_PAGE
VIEW_TOP_REPRESENTATIVES	Ver uma lista com os representantes de vendas com maior número de vendas.	ADMIN_PAGE
VIEW_STATISTICS	Ver estatísticas de vendas, como volume total de vendas ou volume de vendas por vendedor.	ADMIN_PAGE
VIEW_PRODUCT_CATALOGUE	Ver a lista de produtos	CAT_PAGE
VIEW_MEETINGS	Consultar reuniões agendadas	HOM_PAGE
VIEW_PROSPECTS_CLIENTS	Consultar lista de potenciais clientes	HOM_PAGE
VIEW_TOP_CLIENTS	Consultar lista dos clientes com mais encomendas	HOM_PAGE
CREATE_CLIENT	Criar um novo cliente	NEW_CLI
VIEW_CLIENT	Consultar informação sobre um cliente, incluindo reuniões agendadas e encomendas efetuadas	CLI_PAGE
EDIT_CLIENT	Alterar informação acerca de um cliente	
CREATE_ORDER	Criar uma nova encomenda, por parte de um cliente	NEW_ORD
VIEW_ORDER	Consultar informação sobre uma encomenda, incluindo estado e produtos associados	ORD_PAGE
CREATE_MEETING	Agendar uma reunião com um cliente	NEW_MEE
VIEW_MEETING	Consultar e editar informação acerca de uma reunião com um cliente, incluindo sugestões de produtos e notas	MEE_PAGE
EDIT_MEETING	Alterar a data de uma reunião	
DELETE_MEETING	Cancelar uma reunião com um cliente	
VIEW_PRODUCT	Consultar informação sobre um produto, incluindo os armazéns que o têm em stock	PRO_PAGE
SEARCH	Procurar produtos e clientes	

3 Implementação

3.1 Interoperabilidade

3.1.1 Clientes

Tabela 12: Tabela do método get_client.

Webservice ID	get_client
Description	Retorna a informação de um cliente específico.
Core Views	CLI_PAGE
Path	<a href="http://<domain>/api/client?id=<clientID>">http://<domain>/api/client?id=<clientID>
Verb	GET
Input	clientID
Output	{ "id": "EFACSA", "nome": "EFACEC", "nome_fiscal": "EFACEC SA", "morada": "Porto", "telefone": 969509655, "contribuinte": 989922455 }

Tabela 13: Tabela do método get_all_clients.

Webservice ID	get_all_clients
Description	Retorna todos os clientes.
Core Views	HOM_PAGE
Path	<a href="http://<domain>/api/clients">http://<domain>/api/clients
Verb	GET
Input	userID
Output	{ "Clientes": [{ "id": "EFACSA", "nome": "EFACEC", "nome_fiscal": "EFACEC SA", "morada": "Porto", "telefone": 969509655, "contribuinte": 989922455 }, { "id": "LIMA", "nome": "Empreendimentos do Lima", "nome_fiscal": "Empreendimentos do Lima, Lda", "morada": "Bairro PSP", "telefone": , "contribuinte": 202075133 }]} }

Tabela 14: Tabela do método add_client.

Webservice ID	add_client
Description	Adiciona um cliente.
Core Views	NEW_CLI
Path	<a href="http://<domain>/api/client">http://<domain>/api/client
Verb	POST
Input	nome, nome_fiscal, email, morada, telefone, contribuinte
Output	{ "resultado": "OK" }

3.1.2 Reuniões

Tabela 15: Tabela do método get_meetings.

Webservice ID	get_meetings
Description	Retorna as reuniões.
Core Views	HOM_PAGE
Path	<a href="http://<domain>/api/meeting">http://<domain>/api/meeting
Verb	GET
Input	userID
Output	{ "Reunioes": [{ "descricao": "Discussão dos novos produtos da empresa.", "data": 29/10/2016, "tempo": 16h00, "notas": "Não esquecer de mencionar a promoção em vigor para a compra de 100 unidades." }, { "descricao": "Discussão de uma parceria com a equipa de marketing.", "data": 30/10/2016, "tempo": 10h00, "notas": "Nome do representante é João Figueiredo." }] }

Tabela 16: Tabela do método add_meeting.

Webservice ID	add_meeting
Description	Adiciona uma reunião.
Core Views	NEW_MEE
Path	<a href="http://<domain>/api/meeting">http://<domain>/api/meeting
Verb	POST
Input	descricao, data, tempo, notas
Output	{ "resultado": "OK" }

Tabela 17: Tabela do método edit_meeting.

Webservice ID	edit_meeting
Description	Edita uma reunião.
Core Views	
Path	<a href="http://<domain>/api/meeting">http://<domain>/api/meeting
Verb	PUT
Input	descricao, data, tempo, notas
Output	{ "resultado": "OK" }

Tabela 18: Tabela do método delete_meeting.

Webservice ID	delete_meeting
Description	Apaga uma reunião.
Core Views	
Path	<a href="http://<domain>/api/meeting?id=<meetingID>">http://<domain>/api/meeting?id=<meetingID>
Verb	DELETE
Input	meetingID
Output	{ "resultado": "OK" }

3.1.3 Armazéns

Tabela 19: Tabela do método get_warehouse.

Webservice ID	get_warehouse
Description	Retorna a informação dos armazéns.
Core Views	PRO_PAGE
Path	<a href="http://<domain>/api/warehouse?id=<warehouseID>">http://<domain>/api/warehouse?id=<warehouseID>
Verb	GET
Input	warehouseID
Output	{ "name": "ARMAZ", "morada": "warehouse location", "contacto": "warehouse contact" }

3.1.4 Transações

Tabela 20: Tabela do método get_transaction.

Webservice ID	get_transaction
Description	Retorna a transação.
Core Views	TRA_PAGE
Path	<a href="http://<domain>/api/transaction?id=<transactionID>">http://<domain>/api/transaction?id=<transactionID>
Verb	GET
Input	transactionID
Output	{ "clienteID": "EFACSA", "repvendaID": "VENDEDOR", "produtos": [{ "produtoID": 243, "preco": 22.3, "iva": 21, "desconto": 0}, ...], "data": "2016-10-09 15:12:23"}

Tabela 21: Tabela do método post_transaction.

Webservice ID	post_transaction
Description	Adiciona uma transação.
Core Views	NEW_TRA
Path	<a href="http://<domain>/api/transaction">http://<domain>/api/transaction
Verb	POST
Input	clienteID, repvendaID, listaProdutos
Output	{ "resultado": "OK" }

3.1.5 Artigos

Tabela 22: Tabela do método get_article.

Webservice ID	get_article
Description	Retorna um artigo.
Core Views	PRO_PAGE
Path	<a href="http://<domain>/api/article?id=<articleID>">http://<domain>/api/article?id=<articleID>
Verb	GET
Input	articleID
Output	<pre>{ "nome": "nome artigo", "descricao": "descricao artigo", "preco_atual": 12.5, "iva_atual": 13, "stock_total": 100, "stock_disponivel": 85 }</pre>

Tabela 23: Tabela do método get_all_articles

Webservice ID	get_all_article
Description	Retorna todos os artigos.
Core Views	CAT_PAGE
Path	<a href="http://<domain>/api/articles">http://<domain>/api/articles
Verb	GET
Input	
Output	<pre>{ "artigos": [{ "nome": "nome artigo", "descricao": "descricao artigo", "preco_atual": 12.5, "iva_atual": 13, "stock_total": 100, "stock_disponivel": 85 }, ...] } }</pre>

Tabela 24: Tabela do método get_all_articles

Webservice ID	get_warehouse_stock
Description	Retorna o stock de um artigo específico num armazém.
Core Views	PRO_PAGE
Path	<a href="http://<domain>/api/warehouse_article?article=<articleID>&warehouse=<warehouseID>">http://<domain>/api/warehouse_article?article=<articleID>&warehouse=<warehouseID>
Verb	GET
Input	artigoID, armazenID
Output	<pre>{ "stock": 100 }</pre>

3.1.6 Administrador

Tabela 25: Tabela do método get_product_stats.

Webservice ID	get_product_stats
Description	Retorna as estatísticas de um produto num certo intervalo de tempo.
Core Views	
Path	<a href="http://<domain>/api/article=<articleID>&time=<timePeriod>">http://<domain>/api/article=<articleID>&time=<timePeriod>
Verb	GET
Input	articleID, time_period
Output	{ "nome": "nome artigo", "vendas": 10000 }

Tabela 26: Tabela do método get_all_product_stats.

Webservice ID	get_all_product_stats
Description	Retorna estatísticas do número de produtos mais vendidos no intervalo de tempo especificado.
Core Views	
Path	<a href="http://<domain>/api/article=<articleID>&time=<timePeriod>&nr=<nrP>">http://<domain>/api/article=<articleID>&time=<timePeriod>&nr=<nrP>
Verb	GET
Input	articleID, time_period, nrProducts
Output	{ "artigos": [{ "nome": "CloroX", "vendas": 10000 }, { "nome": "Flit", "vendas": 5000 }] }

Tabela 27: Tabela do método get_salesRep_stats.

Webservice ID	get_salesRep_stats
Description	Retorna estatísticas de vendas de um vendedor específico.
Core Views	
Path	<a href="http://<domain>/api/salesRep=<salesRepID>&time=<timePeriod>">http://<domain>/api/salesRep=<salesRepID>&time=<timePeriod>
Verb	GET
Input	salesRepID, time_period
Output	{ "nome": "João Claro", "vendas": 10000 }

Tabela 28: Tabela do método get_all_salesRep_stats.

Webservice ID	get_all_salesRep_stats
Description	Retorna estatísticas de vendas do número de vendedores especificado.
Core Views	
Path	<a href="http://<domain>/api/salesRep=<salesRepID>&time=<timePeriod>&nr=<nrS>">http://<domain>/api/salesRep=<salesRepID>&time=<timePeriod>&nr=<nrS>
Verb	GET
Input	salesRepID, time_period, nrSalesRep
Output	<pre>{ "vendedores": [{ "nome": "João Claro", "vendas": 10000 }, { "nome": "Joana Cabral", "vendas": 3000 }] }</pre>

3.2 Caminhos

4 Arquitetura

estrutura lógica, etc

5 Especificação vs Implementação

«Project Specifications vs Delivered Project»

6 Conclusões

«Lessons Learned» Finda esta fase, consideram-se atingidos os objetivos definidos para esta primeira fase: foi feita a descrição do projeto – uma aplicação web que oferecerá funcionalidades de uma solução SFA. Enumeraram-se as funcionalidades a implementar e identificaram-se os pontos de ligação com o ERP Primavera.