



Universidade do Porto

Faculdade de Engenharia

**FEUP**

SISTEMAS DE INFORMAÇÃO

4º ANO DO MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA E  
COMPUTAÇÃO

---

# *Sales Representative Automation*

Especificação Funcional

---

## *Authors:*

Filipa RAMOS

- up201305378@fe.up.pt

Gil DOMINGUES

- up201304646@fe.up.pt

Pedro PONTES

- up201305367@fe.up.pt

Pedro MELO

- up201305618@fe.up.pt

21 de Outubro de 2016

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Especificação</b>	<b>3</b>
2.1	Descrição . . . . .	3
2.2	Funcionalidades . . . . .	3
<b>3</b>	<b>Core Views</b>	<b>4</b>
3.1	Client Page: Display of client information and transaction history . . . . .	4
3.2	Homepage: Overview of schedule . . . . .	5
3.3	Meeting Page: Overview of meeting . . . . .	6
3.4	New Client Page: Create page for new potencial client . . . . .	7
3.5	New Meeting Page: Schedule meeting with client . . . . .	8
3.6	New Order Page: Sell products to client . . . . .	9
3.7	Order Status Page: Overview of order status . . . . .	10
3.8	Product Page: Overview of product . . . . .	11
3.9	Catalog Page . . . . .	12
3.10	Administration Page . . . . .	13
<b>4</b>	<b>Interoperabilidade</b>	<b>14</b>
4.1	Clientes . . . . .	14
4.2	Reuniões . . . . .	15
4.3	Armazéns . . . . .	16
4.4	Transações . . . . .	17
4.5	Artigos . . . . .	18
4.6	Administrador . . . . .	19
<b>5</b>	<b>Conclusão</b>	<b>20</b>

## 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 fará a comunicação com o Primavera, passando a informação à aplicação front-end através de uma API RESTful.

### 2.2 Funcionalidades

Tabela 1: Descrição das funcionalidades do software.

Funcionalidade	Finalidade	CoreView
REGISTER_REPRESENTATIVE	Registar 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 Core Views

#### 3.1 Client Page: Display of client information and transaction history

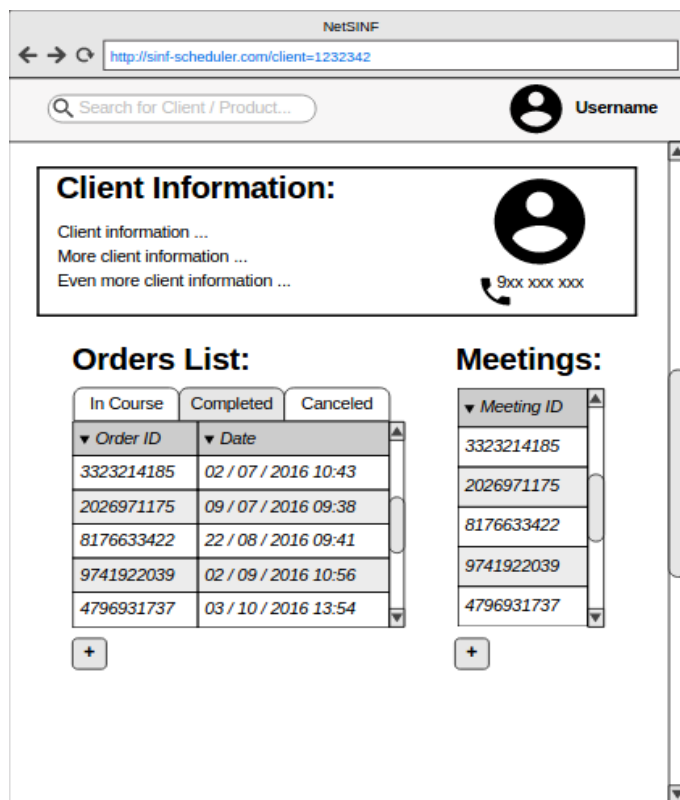


Figura 1: Core View da página de cliente.

Tabela 2: Core View Client Page especificação.

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

## 3.2 Homepage: Overview of schedule

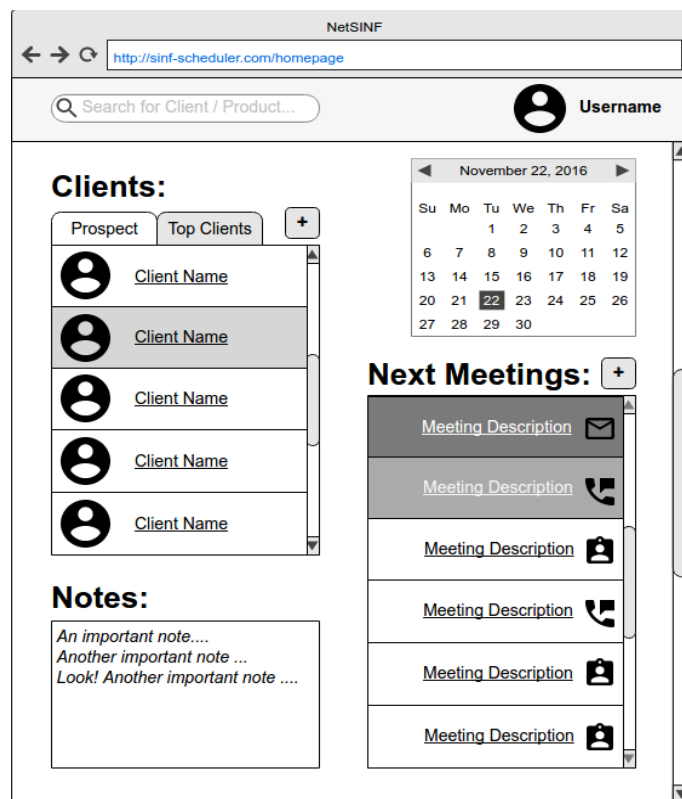


Figura 2: Core View da página inicial.

Tabela 3: Core View Homepage especificação.

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

### 3.3 Meeting Page: Overview of meeting

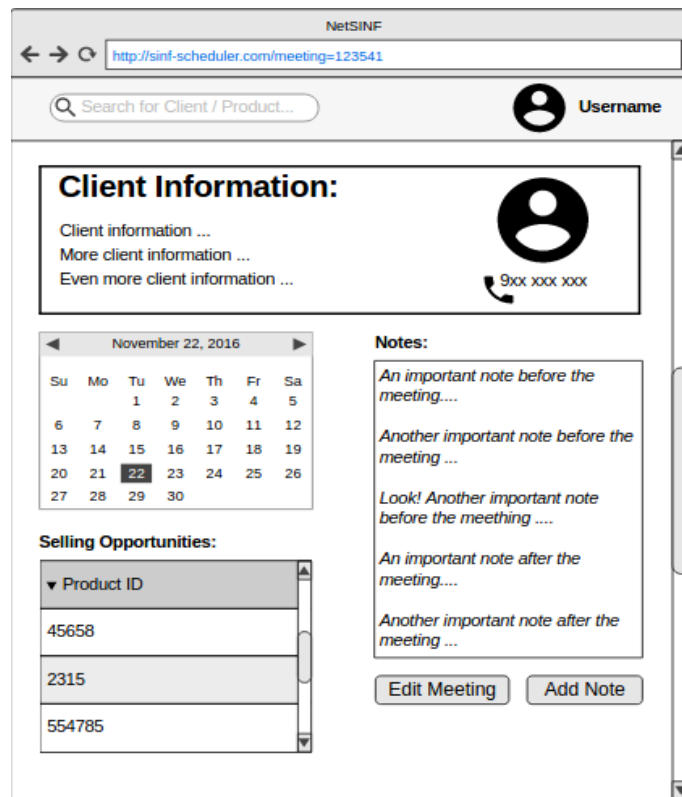


Figura 3: Core View da página das reuniões agendadas.

Tabela 4: Core View Meeting Page especificação.

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

3.4 New Client Page: Create page for new potencial client

NetSINF

← → ↻

Username

**Create New Client:**

Name:

Address:

Tax Name:

NIF:

Email:

Phone:

+

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

Tabela 5: Core View New Client especificação.

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

### 3.5 New Meeting Page: Schedule meeting with client

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

Tabela 6: Core View New Meeting Page especificação.

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



### 3.6 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 7: Core View New Order Page especificação.

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

### 3.7 Order Status Page: Overview of order status

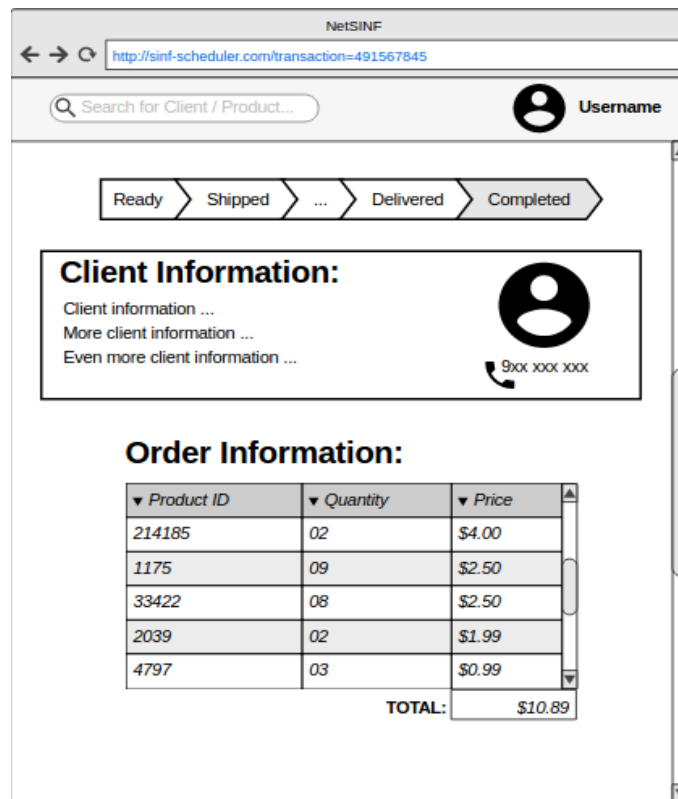


Figura 7: Core View da página onde se visualiza um pedido de um cliente.

Tabela 8: Core View Order Status Page especificação.

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

### 3.8 Product Page: Overview of product

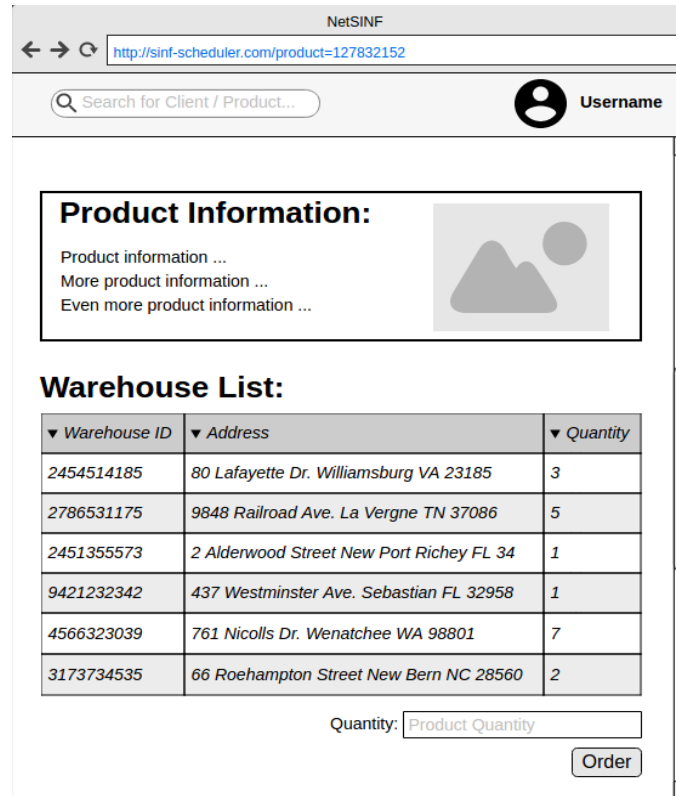


Figura 8: Core View da página de um produto e a sua disponibilidade em cada armazém.

Tabela 9: Core View Product Page especificação.

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

### 3.9 Catalog Page

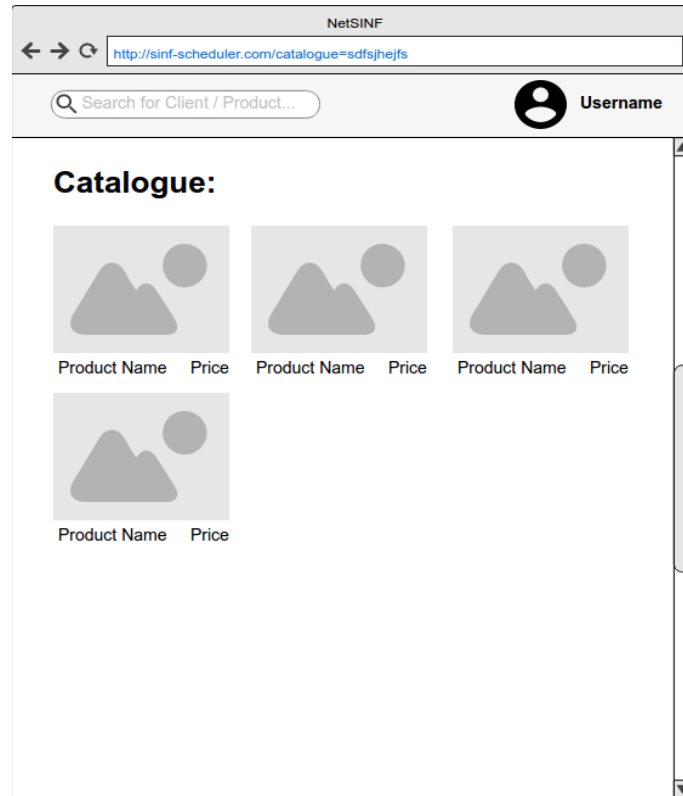


Figura 9: Core View da \_\_\_\_\_

3.10 Administration Page

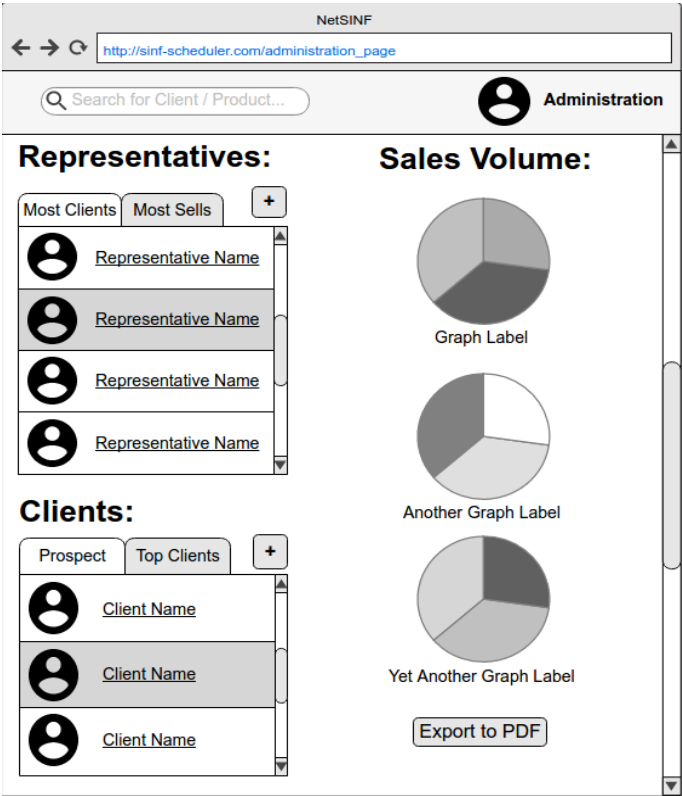


Figura 10: Core View da \_\_\_\_\_

\_\_\_\_\_

## 4 Interoperabilidade

### 4.1 Clientes

Tabela 10: Tabela do método get\_client.

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

Tabela 11: Tabela do método get\_all\_clients.

<b>Webservice ID</b>	get_all_clients
<b>Description</b>	Retorna todos os clientes.
<b>Core Views</b>	HOM_PAGE
<b>Path</b>	<a href="http://&lt;domain&gt;/api/clients">http://&lt;domain&gt;/api/clients</a>
<b>Verb</b>	GET
<b>Input</b>	userID
<b>Output</b>	<pre>{ "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   } ]}</pre>

Tabela 12: Tabela do método add\_client.

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

## 4.2 Reuniões

Tabela 13: Tabela do método get\_meetings.

<b>Webservice ID</b>	get_meetings
<b>Description</b>	Retorna as reuniões.
<b>Core Views</b>	HOM_PAGE
<b>Path</b>	<a href="http://&lt;domain&gt;/api/meeting">http://&lt;domain&gt;/api/meeting</a>
<b>Verb</b>	GET
<b>Input</b>	userID
<b>Output</b>	{ "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 14: Tabela do método add\_meeting.

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

Tabela 15: Tabela do método edit\_meeting.

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

Tabela 16: Tabela do método delete\_meeting.

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

### 4.3 Armazéns

Tabela 17: Tabela do método get\_warehouse.

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



## 4.4 Transações

Tabela 18: Tabela do método get\_transaction.

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

Tabela 19: Tabela do método post\_transaction.

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

## 4.5 Artigos

Tabela 20: Tabela do método get\_article.

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

Tabela 21: Tabela do método get\_all\_articles

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

Tabela 22: Tabela do método get\_all\_articles

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

## 4.6 Administrador

Tabela 23: Tabela do método get\_product\_stats.

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

Tabela 24: Tabela do método get\_all\_product\_stats.

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

Tabela 25: Tabela do método get\_salesRep\_stats.

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

Tabela 26: Tabela do método get\_all\_salesRep\_stats.

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

## 5 Conclusão

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.

## Referências