

Online/Web Shop

Interoperability



Mestrado Integrado em Engenharia Informática e
Computação

Sistemas de Informação

Grupo AC:

Beatriz de Henriques Martins - up201502858@fe.up.pt
Bruno Miguel de Sousa Pinto - up201502960@fe.up.pt
Nádia de Sousa Varela de Carvalho - ei12047@fe.up.pt
Rúben José da Silva Torres - up201405612@fe.up.pt
Tiago Lascasas dos Santos - up201503616@fe.up.pt

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

December 23, 2018

1 Overview

Our project is an online bookshop where people can search for books, see their information and buy them. Users are able to use advanced search capabilities, browse books by categories and buy one or more books in a single transaction using various payment methods.

The system is mainly based on the Book Depository, which is an online book retailer for physical books, and the type of commerce to practiced is the business to client, which replaces the physical processes that involve commercial transactions by performing them remotely via the Web.

On this project, we used Enterprise Resource Planning software Primavera, which was adapted to deal with all the data and processes surrounding inventories, transactions and information about each book. The application's back-end and the communication layer with the ERP were made in C# using the ASP.NET framework, and the front-end using Bootstrap. Finally, GitHub was used for version control and general development.

2 Core Views

In this section we show and explain each core view in relation to the final interfaces in which they appear. The core views are always surrounded by a **dark blue rectangle**, as the core view seldom matches the whole page. For each element of each core view, we will also show which Primavera Web Service (WS) implements that feature.

2.1 Homepage/Splash page

2.1.1 User and Business Goals:

1. Present the website to the user
2. Authenticate/Register on the website
3. Search for a Book
4. See popular/recent books
5. Go to a category page
6. Go to user's cart page

2.1.2 Core View:

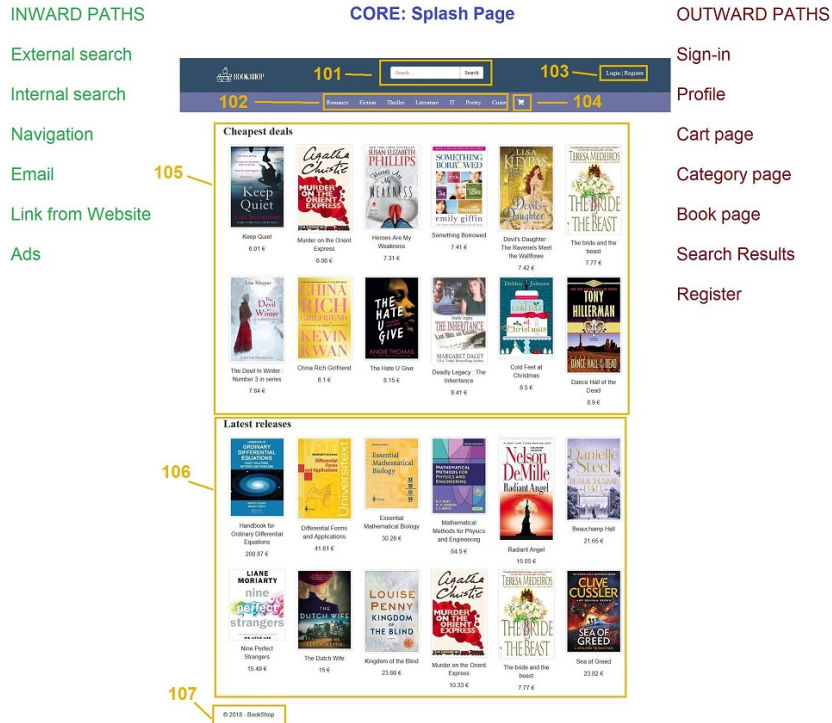


Figure 1: Core View: V.001: splash page of the website

2.1.3 Elements of the core (in yellow on core view above) :

1. **101:** Search Bar
2. **102:** Navigation "by Categories" Bar
3. **103:** Profile Section (if not logged in: Login and Register "buttons" for modals, else Profile, Orders and Logout "buttons")
4. **104:** Cart Button on Navigation Bar
5. **105:** Cheapest Deals
6. **106:** Latest Releases
7. **107:** Footer

2.1.4 Primavera Web Services Used for each element of the core:

1. **for 101:** WS06
2. **for 102:** WS05
3. **for 103:** WS01, WS02, WS03
4. **for 105:** WS07
5. **for 106:** WS08

2.2 Page for each book

2.2.1 User and Business Goals:

1. See all the information about the Product
2. Allow the user to add the Book to the cart

INWARD PATHS

External search

Internal search

Navigation

Email

Link from Website

CORE: Book page

OUTWARD PATHS

Add to cart



Figure 2: Core View: V_002: Product(Book) page

2.2.2 Elements of the core (in yellow on core view above) :

1. **201:** Image of the Product
2. **202:** Specification of the Product
3. **203:** Add to Cart Button

2.2.3 Primavera Web Services Used for each element of the core:

1. for **202:** WS04

2.3 Page for each category

2.3.1 User and Business Goals:

1. Search books by category

INWARD PATHS

External search

Internal search

Navigation

Email

Link from Website

CORE: Category page

OUTWARD PATHS

Book page

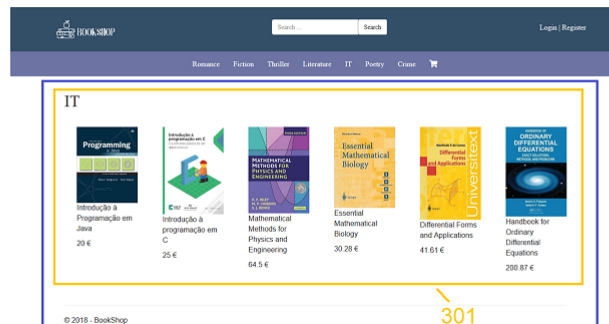


Figure 3: Core View: V_003: Category(Book) page

2.3.2 Elements of the core (in yellow on core view above) :

1. **301:** Grid showing books from the chosen category

2.3.3 Primavera Web Services Used for each element of the core:

1. for **301:** WS05

2.4 Page for search results

2.4.1 User and Business Goals:

1. See the Product whose title relates to the search made by the user

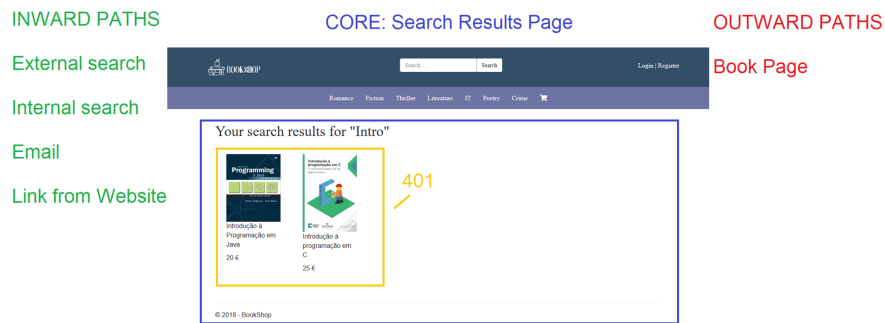


Figure 4: Core View: V_004: Search Results page

2.4.2 Elements of the core (in yellow on core view above) :

1. **401:** Grid showing the books that relate to the search made by the user (partial search by title)

2.4.3 Primavera Web Services Used for each element of the core:

1. for **401:** WS06

2.5 Orders page

2.5.1 User and Business Goals:

1. See orders and their status and specifications

INWARD PATHS

External search

Navigation

Email

CORE: Orders Page

OUTWARD PATHS

Order Page

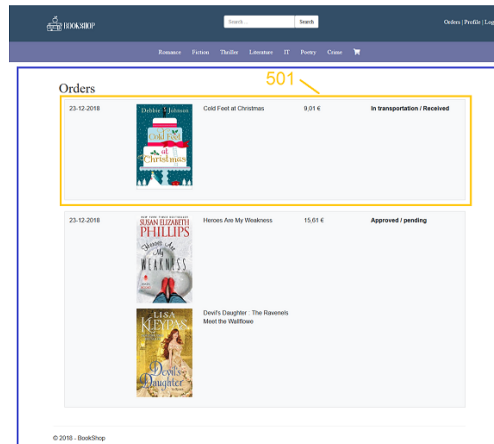


Figure 5: Core View: V_005: Orders page

2.5.2 Elements of the core (in yellow on core view above) :

1. **501:** Element holding the information about an individual order, with the total price, date, the book(s) chosen and the order state.

2.5.3 Primavera Web Services Used for each element of the core:

1. for **501:** WS11, WS12

2.6 Specific Order page

2.6.1 User and Business Goals:

1. See a specific order and all information related to it
2. Access receipt of the order

INWARD PATHS

External search

Navigation

Email

Orders Page

CORE: Order Page

OUTWARD PATHS

Splash Page

Order 2

Info order

Name	Tiago Santos
Address	Rua Eng. Fátima de Almeida, 313, 4º esq
Postal code	4510-260
NIF	248699623
Country	PT

Items

Name	Quant.	s/IVA	c/IVA
Susan Elizabeth Phillips - Heroes Are My Weakness	1 x	7.31 €	7.74 €
Lisa Kleypas - Devil's Daughter	1 x	7.42 €	7.87 €

Info about order

Order nr	2
State	Approved / pending

Total items

Total items	15.61 €
Shipping	0.00 €
TOTAL	15.61 €

© 2018 - BookShop

Figure 6: Core View: V_006: Specific Order page

2.6.2 Elements of the core (in yellow on core view above) :

1. **601:** Book(s) chosen for the order
2. **602:** User related information
3. **603:** Cost information
4. **604:** Order information

2.6.3 Primavera Web Services Used for each element of the core:

1. for **601:** WS11
2. for **602:** WS02, WS11
3. for **603:** WS11
4. for **604:** WS11

2.7 Cart page

2.7.1 User and Business Goals:

1. Review all the order details
2. Make an order

INWARD PATHS

External search

Navigation

Email

CORE: Cart Page

OUTWARD PATHS

Splash Page

The screenshot shows the 'Cart' page of a web application. The page has a dark blue header with the 'BookShop' logo and navigation links: Home, About, Contact, My Account, Cart, and Checkout. A search bar is located in the top right. The main content area is titled 'Cart' and contains three sections highlighted with yellow boxes and numbered annotations:

- 701:** 'Info order and Payment' form. It includes fields for Name (Rafaela Santos), Address (Rua das Flores, 123, 12345-678), P. code (4567-890), NF (123456789), and Phone (11 123456789). A link below says 'To edit this information, please head over to your profile page'.
- 702:** Order summary table. It lists items with their names, quantities, and prices.
- 703:** A green 'Checkout' button at the bottom right.

Name	#	with	with
SOMETHING BORR' WED	2	7.41 €	7.85 €
Only Filthy Rich Love	1	13.01 €	13.79 €

Figure 7: Core View: V_007: Cart Page

2.7.2 Elements of the core (in yellow on core view above) :

1. **701:** Order details form
2. **702:** Order summary
3. **703:** Place the order Button

2.7.3 Primavera Web Services Used for each element of the core:

1. for **702:** WS04
2. for **703:** WS09

2.8 Profile page

2.8.1 User and Business Goals:

1. Access my profile information
2. Change profile information

INWARD PATHS

Navigation

CORE: Profile Page

OUTWARD PATHS

Splash Page

Profile	
User ID	Tiago
Name	Tiago Santos
NIF	64287048860
Address	Rua Eng. Fátima de Almeida, 313, 4º esq
Postal code	4510-260
Phone	224633959
Email	emil123ewmpio@gmail.com
Password	*****

Figure 8: Core View: V_008: Profile Page

2.8.2 Elements of the core (in yellow on core view above) :

1. **801:** Information of Profile
2. **802:** Update information button

2.8.3 Primavera Web Services Used for each element of the core:

1. for **801:** WS02
2. for **802:** WS02, WS03

3 Other features

We consider most of our features to be core views. The only feature that we consider non-essential to the project is authenticating and registering a new user. They both are implemented using modals available in every page, and as such lack a page of their own. Still, we present those modals here and establish which Web Services implements them.

3.1 Login modal

3.1.1 User and Business Goals:

1. Authenticate an existing user

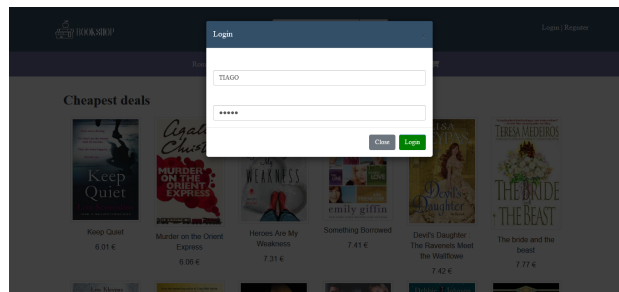


Figure 9: Minor functionality MF_001: Login Modal

3.1.2 Primavera Web Services used:

- WS02
- WS03

3.2 Register modal

3.2.1 User and Business Goals:

1. Register a new user

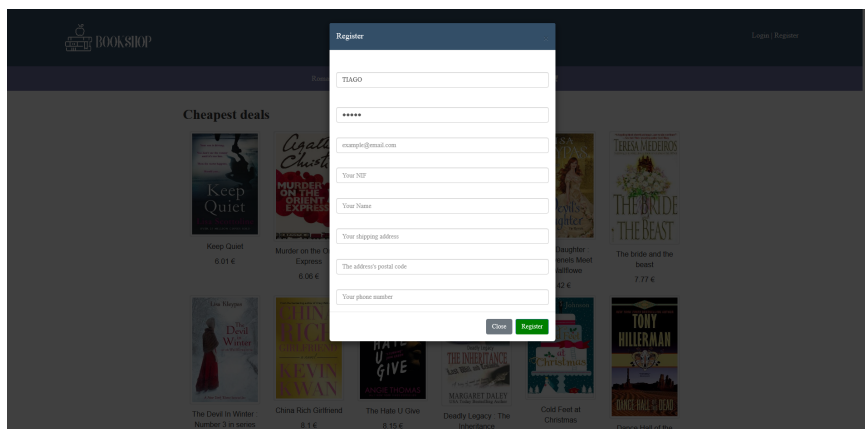


Figure 10: Minor functionality MF_002: Register Modal

3.2.2 Primavera Web Services used:

- WS01
- WS02
- WS03

4 Web Services

We have developed 12 distinct Web Services by adapting the provided Web API. Comparing to the previous deliverable, we have added a new service, WS12, as it was required in order to find the ID of an existing document. WS10, however, was deprecated, as it was made irrelevant with the revised document flow that we adopted for this final delivery. **At the end of this report**, you can find a description of each Web Service, including its route, purpose and an example of an input and expected output.

5 Sitemap

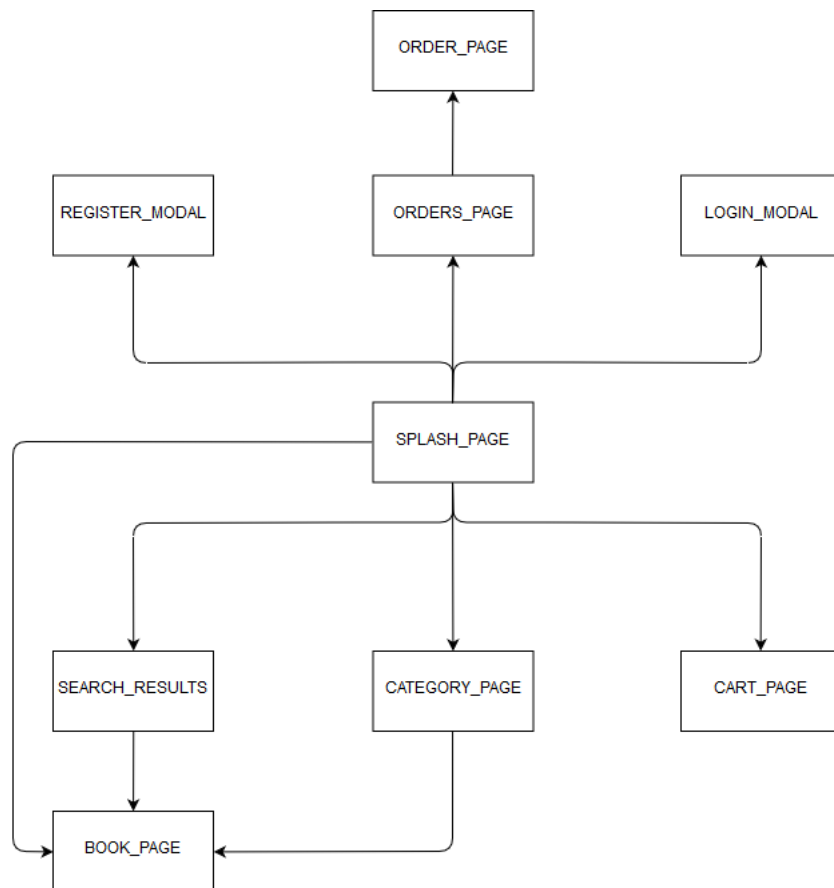


Figure 11: Sitemap

6 System Architecture

6.1 Architecture Overview

Our system architecture is composed of three main components:

- **Website:** it uses ASP.NET Core for the backend, and Bootstrap for the frontend. Its only purpose is to render and provide pages, as all business logic is done by Primavera. The only thing it stores is authentication and cart variables for each user session.
- **API:** the API makes the bridge between the Web backend and Primavera. It provides the aforementioned Web Services, and requires authentication via tokens. The only entity authenticated is the Web backend, and all client requests are forwarded through a single connection.
- **Primavera:** the ERP provides all the functionalities required to implement the business objectives, such as managing stocks, clients and purchases.

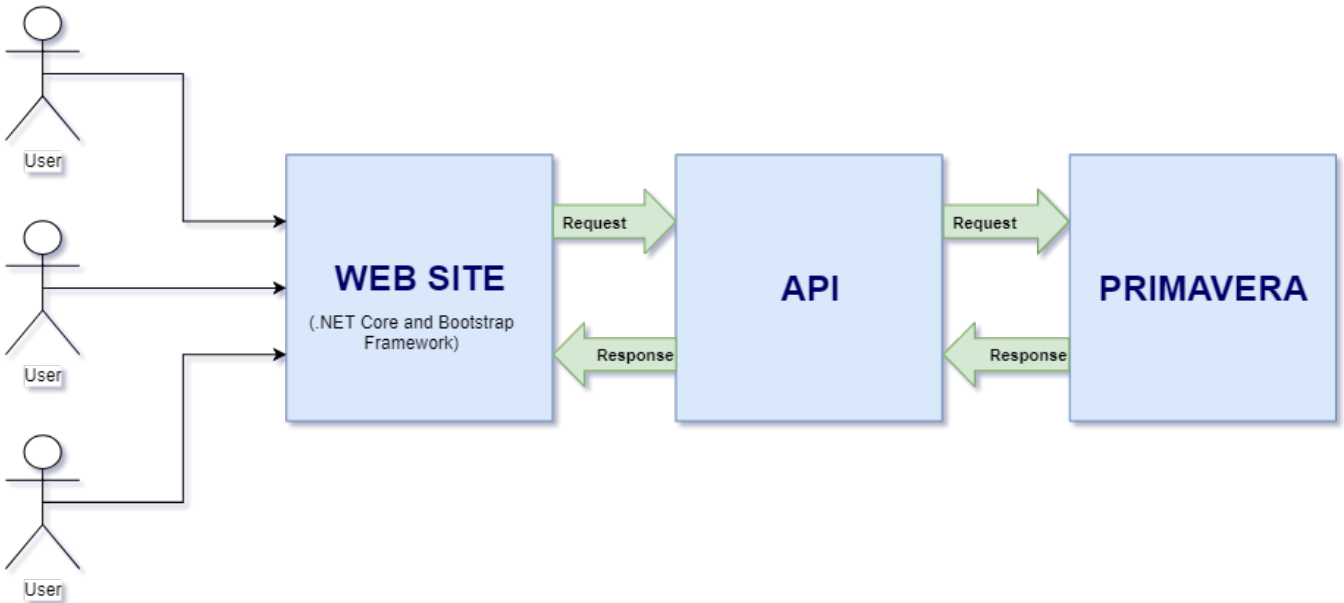


Figure 12: System Architecture

6.2 Office backend and document flow on Primavera

After an Order is made by an user (ECL) on the website the rest of the document flow should be done on the Primavera back office in order to finalize an order:



Figure 13: Order document flow

7 Project Specifications vs Delivered Project

We believe that we have implemented all the features we had presented in the Project Specifications, and that we accomplished all the objectives mentioned in the project themes presentation.

All core views and user stories proposed on the first deliverable were implemented successfully, with no additional functionalities added or removed.

The Web Services proposed on the second deliverable were also used for their respective tasks, and an additional WS was created to fetch information about created documents.

The project planning established on the first deliverable was followed. We took longer than we predicted to get acquainted with Primavera and to pick and configure the right API calls, but the postponing of the second deliverable date helped to solve that. Despite not having two minor functionalities done by the time of the presentation (the login and register), we implemented them immediately after. We also solved the small misinterpretation of the IVA values returned from Primavera that was pointed out during the presentation.

The only lacking part of our project is about the warehouse management, as we always take the items from a default warehouse rather than from the one that is closer or with larger stocks. This is due to the limiting capabilities of the API given to us, which did not let us do warehouse and stock management the way we intended to. The few capabilities it provides do not fall within the scope of our project either.

8 Lessons Learned

On this project, we learned how to integrate an e-commerce platform, such as our book shop, with an ERP, in this instance Primavera.

It was challenging, as we have decided to work with ASP.NET, a web development framework different from the others we had used in the past and with a programming language with we had no experience (C#). Also, there were many problems with the material that we were given for working on the project, during its development, as it was either confusing, delivered way too late or filled with bugs. In fact, the decision to adopt ASP.NET in the first place came from the erroneous assumption that we would have to extend Primavera in order to integrate it with our Web application. Since Primavera can be extended through C#, we thought that making the Web backend in that language would provide a seamless integration with Primavera, but given that the communication with Primavera was done instead through the provided Web API, ASP.NET's advantages ended up being irrelevant.

While we were working on this project, we were able to learn more and more about Primavera and ERPs in general, in particular the flow of documents in order to implement a certain business objective (in our case, making online purchases with home delivery). We also discovered the hardships of extending an ERP in order to fit the particular needs of a project, as we had to deal with creating new user fields and create our own queries in order to go beyond the provided capabilities of the provided Web API.

9 Annex: Web Services Documentation

Web Service	WS01
Description	Creates a new user/client
Related Core View s	V_001
Route	Base/Clientes/Atualiza
Input example	<pre>{ "Cliente": "1", "Nome": "Manfred Reubens", "Morada": "Praceta Conde Arnoso 60", "CodigoPostal": "2640-467", "Telefone": "6705966965", "NumContribuinte": "64287048860", "Pais": "PT", "Moeda": "EUR" }</pre>
Expected Output	HTTP 204

Web Service	WS02
Description	Gets the relevant information of an user/client
Related Core Views	V_001, V_006, V_008
Route	Base/Clientes/Edita/{cliente_id}
Input example	<pre>cliente_id = "TIAGO" (parameter in the route)</pre>
Expected Output	<pre>{ "CodigoTabLog": "Cliente", "ChaveLog": "Cliente", "EstadoBE": "", "Cliente": "TIAGO", "Nome": "Tiago Santos", "Descricao": "", "Morada": "Rua Eng. Farinas de Almeida, 313, 4º esq", (...) "CamposUtil": [{ "Conteudo": "ValorValoremailxyz@gmail.com", "Nome": "CDU_email", "Valor": "emailxyz@gmail.com", "Objecto": null, "Tipo": 1, "ChaveLog": "Nome", "EstadoBE": "", "TipoSimplificado": 1 }, { "Conteudo": "ValorValor03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4", "Nome": "CDU_password_hash", "Valor": "03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4", "Objecto": null, </pre>

	<pre> "Tipo": 1, "ChaveLog": "Nome", "EstadoBE": "", "TipoSimplificado": 1 }], (...) } </pre>
<div> <div></div> <div></div> </div>	

Web Service	WS03
Description	Updates the information of an user/client
Related Core Views	V_008
Route	Base/Clientes/Atualiza
Input example	<pre> //Input is the same as the output of WS02, but with some changed attributes, //such as Morada, CDU_email and CDU_password_hash { "CodigoTabLog": "Cliente", "ChaveLog": "Cliente", "EstadoBE": "", "Cliente": "TIAGO", "Nome": "Tiago Santos", "Descricao": "", "Morada": "Nova Morada, 451", (...) "CamposUtil": [(...) { "Conteudo": "ValorValornewemail@gmail.com", "Nome": "CDU_email", "Valor": "newemail@gmail.com", "Objecto": null, "Tipo": 1, "ChaveLog": "Nome", "EstadoBE": "", "TipoSimplificado": 1 }, { "Conteudo": "ValorValor253C2E786C2414DCAEC8DBF11DF515B5075371454B93A5687D24D96DDBF3B939", "Nome": "CDU_password_hash", "Valor": "253C2E786C2414DCAEC8DBF11DF515B5075371454B93A5687D24D96DDBF3B939", "Objecto": null, "Tipo": 1, "ChaveLog": "Nome", "EstadoBE": "", "TipoSimplificado": 1 }], (...) } </pre>
Expected Output	HTTP 204
<div> <div></div> <div></div> </div>	

Web Service	WS04
Description	Gets all the relevant information of a book
Related Core Views	V_004, V_005, V_006, V_007
Route	Administrador/Consulta
Input example	<pre>-- select the relevant info of book "B0001" Select Descricao, CDU_Autor, CDU_Sinopse, CDU_ISBN, CDU_Editora, CDU_Capa, CDU_Paginas, CDU_Ano, CDU_Dimensoes, Familia, CDU_Idioma, PVP1 from Artigo, ArtigoMoeda where Artigo.Artigo = 'B0001' and Artigo.Artigo = ArtigoMoeda.Artigo</pre>
Expected Output	<pre>{ "DataSet": { "Table": [{ "Descricao": "A Brief History Of Time", "CDU_Autor": "Stephen Hawking", "CDU_Sinopse": "A landmark volume in science writing by one of the great minds of our t "CDU_ISBN": "0857501003", "CDU_Editora": "Transworld Publishers Ltd", "CDU_Capa": "Paperback", "CDU_Paginas": 272, "CDU_Ano": "05 Feb 2015", "CDU_Dimensoes": "127 x 198 x 17mm", "Familia": "FICTION", "CDU_Idioma": English, "PVP1": 25.99 }] }, "Query": "System.Data.SqlClient.SqlCommand" }</pre>

Web Service	WS05
Description	Gets a set of books that belong to the specified category
Related Core Views	V_001, V_004
Route	Administrador/Consulta
Input example	<pre>-- select all books that belong to the category "IT" Select Artigo.Artigo, Descricao, PVP1, CDU_Autor from Artigo, ArtigoMoeda where Familia = 'IT' and Artigo.Artigo = ArtigoMoeda.Artigo</pre>
	<pre>{</pre>

Expected Output	<pre> "DataSet": { "Table": [{ "Artigo": "B0003", "Descricao": "Introdução à Programação em Java", "PVP1": 20, "CDU_Autor": "António Manuel Adrego da Rocha" }, { "Artigo": "B0004", "Descricao": "Introdução à programação em C", "PVP1": 25, "CDU_Autor": "António Manuel Adrego da Rocha" }] }, "Query": "System.Data.SqlClient.SqlCommand" } </pre>
-----------------	---

Web Service	WS06
Description	Searches books by title, including partial strings
Related Core View s	V_006, V_001
Route	Administrador/Consulta
Input example	<pre> -- select all books with "Intro" in the title "Select Artigo.Artigo, Descricao, CDU_Autor, PVP1 from Artigo, ArtigoMoeda where Descricao like '%Intro%' and Artigo.Artigo = ArtigoMoeda.Artigo" </pre>
Expected Output	<pre> { "DataSet": { "Table": [{ "Artigo": "B0003", "Descricao": "Introdução à Programação em Java", "CDU_Autor": "António Manuel Adrego da Rocha", "PVP1": 20 }, { "Artigo": "B0004", "Descricao": "Introdução à programação em C", "CDU_Autor": "António Manuel Adrego da Rocha", "PVP1": 25 }] }, "Query": "System.Data.SqlClient.SqlCommand" } </pre>

Web Service	WS07
Description	Gets a list of books ordered by smallest price
Related Core View s	V_001

Route	Administrador/Consulta
Input example	<pre>Select Artigo.Artigo, Descricao, PVP1, CDU_Autor from Artigo, ArtigoMoeda where Artigo.Artigo = ArtigoMoeda.Artigo order by PVP1</pre>
Expected Output	<pre>{ "DataSet": { "Table": [{ "Artigo": "B0026", "Descricao": "Keep Quiet", "PVP1": 6.01, "CDU_Autor": "Lisa Scottoline", "STKActual": 0 }, { "Artigo": "B0024", "Descricao": "Murder on the Orient Express", "PVP1": 6.06, "CDU_Autor": "Agatha Christie", "STKActual": 0 }, { "Artigo": "B0022", "Descricao": "Heroes Are My Weakness", "PVP1": 7.31, "CDU_Autor": "Susan Elizabeth Phillips", "STKActual": 0 }, (...)] }, "Query": "System.Data.SqlClient.SqlCommand" }</pre>

Web Service	WS08
Description	Gets a list of books ordered by the last date they were updated (first is the most recent)
Related Core Views	V_008
Route	Administrador/Consulta
Input example	<pre>Select Artigo.Artigo, Descricao, PVP1, CDU_Autor, DataUltimaAtualizacao from Artigo, ArtigoMoeda where Artigo.Artigo = ArtigoMoeda.Artigo order by DataUltimaAtualizacao desc</pre>
	<pre>{ "DataSet": { "Table": [{ "Artigo": "B0003", "Descricao": "Introdução à Programação em Java", "PVP1": 20, "CDU_Autor": "António Manuel Adrego da Rocha", "DataUltimaAtualizacao": "2018-12-02T02:04:21" }, {</pre>

Expected Output	<pre> "Artigo": "B0005", "Descricao": "Foi Sem Querer Que Te Quis", "PVP1": 12, "CDU_Autor": "Raul Minh'alma ", "DataUltimaAtualizacao": "2018-12-02T02:03:39" }, { "Artigo": "B0004", "Descricao": "Introdução à programação em C", "PVP1": 25, "CDU_Autor": "António Manuel Adrego da Rocha", "DataUltimaAtualizacao": "2018-12-02T01:59:40" }, { "Artigo": "B0002", "Descricao": "Romeu e Julieta", "PVP1": 25, "CDU_Autor": "William Shakespeare ", "DataUltimaAtualizacao": "2018-12-02T01:50:31" }, { "Artigo": "B0001", "Descricao": "A Brief History Of Time", "PVP1": 25.99, "CDU_Autor": "Stephen Hawking", "DataUltimaAtualizacao": "2018-12-02T01:46:43" }, (...)] }, "Query": "System.Data.SqlClient.SqlCommand" } </pre>
-----------------	---

Web Service	WS09
Description	Makes a purchase of the specified items, producing an ECL document on the ERP
Related Core View s	V_007
Route	Administrador/Consulta
Input example	<pre> { "Linhas": [{ "Artigo": "B0001", "Quantidade": "1" }, { "Artigo": "B0002", "Quantidade": "1" }], "Tipodoc": "ECL", "Serie": "A", "Entidade": "C0001", "TipoEntidade": "C", "DataDoc": "12/11/2018", "DataVenc": "12/12/2018" } </pre>
Expected Output	true

(WS10 has become deprecated in the final version of the project, as it turned out to not be needed)

Web Service	WS11
Description	Gets the specified ECL document
Related Core Views	V_005, V_006
Route	Vendas/Docs/Edita/000/ECL/A/{id}
Input example	id = 1 (parâmetros da rota)
Expected Output	<pre> { "IDDiarioCaixa": "", "OrigemPOS": false, (...) "PropExcluirLog": "", "Linhas": [{ "ChaveLog": "Artigo,Armazem,Lote,TipoLinha", "EstadoBE": "", (...) "IntrastatValorLiq": 0, "NumLinDocOriginal": 0, "CamposUtil": [{ "Conteudo": "ValorValor", "Nome": "CDU_LinVar1", (...) "EstadoBE": "", "TipoSimplificado": 1 }, (...)], (...) "BaseIncidencia": 0, "RegraCalculoIncidencia": 0, "DadosAdiantamento": { "Conteudo": "DifArredondamentoMAlt0DifArredondamentoMBase0DifCambioMAlt0DifCambioMBase0", "MoedaDocOrig": "", (...) }, "DadosImpostoSelo": { "Ano": 0, "Selo": "", (...) }, "ValorLiquidoDesconto": 0, "IvaValorDesconto": 0, (...) "ValorProcessoAnosSeguintes": 0, "AbvtAp1": "VND" }], "Retencoes": [], "ResumoIva": [{ "ChaveLog": "Filial,Serie,Tipodoc,NumDoc,CodIva", "EstadoBE": "", (...) "ValorDesconto": 0, "AbvtAp1": "BAS" }], "ID": "8cf01dd8-f667-11e8-b6a5-08002756714a", "IDCabecMovCb1": "a5f9744c-f667-11e8-b6a5-08002756714a", (...) "ValorEntregueCambio": 0, </pre>

```
"EmModoEdicao": true,
"CamposUtil": [
  {
    "Conteudo": "ValorValor",
    "Nome": "CDU_CabVar1",
    (...)
    "EstadoBE": "",
    "TipoSimplificado": 1
  },
  (...)
],
(...)
"ContratoFactoring": "",
"ATDocCodeID": "",
"CargaDescarga": {
  (...)
  "ATDocCodeID": "",
  "AbvtApl": "BAS"
},
"EntidadeDescarga": "C0001",
"LocalDescarga": "V/ Morada",
(...)
"HoraDefinida": false,
"AbvtApl": "VND"
}
```

Web Service	WS12
Description	Gets information about the ECL documents
Related Core View s	V_005
Route	Base/Series/ListaSeries/V/ECL/false
Input example	<pre>[{ "CodigoTabLog": "Series", "ChaveLog": "Documento,Serie", "EstadoBE": "", "Documento": "ECL", "Serie": "A", "Ordem": 1, "Descricao": "Série A", "SerieInactiva": false, (...) "Interna": false, "eGarAbreDocumento": false, "eGAR_Comunica": false, "AbvtApl": "BAS" }]</pre>
Expected Output	HTTP 200