P1 - Online/Web Store

Master in Informatics and Computing Engineering

Information Systems - FINAL DOCUMENT

Turma 3 - Grupo T :

201506365 - Daniel Pereira Machado - up201506365@fe.up.pt

201504875 - João Filipe Lopes de Carvalho - up201504875@fe.up.pt

201506265 - Nuno Miguel Outeiro Pereira - up201506265@fe.up.pt

201504570 - Sofia Catarina Bahamonde Alves - up201504570@fe.up-pt

Faculty of Engineering of the University of Porto

Rua Roberto Frias, sn, 4200-465 Porto, Portugal

December 22, 2018

Table of Contents

Overview

This project consists in the development of a web application for the management of an online **technology store**. This application allows the user to make an order registration, consult purchase history and track the status of an order. Besides that, every product has a score given by the users. The users of this website are separated in two groups "Administrators" and regular "Users". Administrators are able to add products, block users and have access to information about the sales, purchases, inventory, billing and other. Regular Users have access to the products pages and can edit their profile information, also they are able to keep items in a shopping cart (even when logged out) and, at the end of their purchases, place a order for one or more items desired.

This application uses ERP Primavera as a backoffice for the sales process and stock management. The communication between the app and the ERP will be done through a Web API provided to facilitate the process. To build our Web app we use a combination of **Handlebars** (Frontend), **Node**, **Express** and **MongoDB** (Backend)**.** This last one is a database platform that allows to access data easily, in which we have stored only crucial information for the users authentication such and email and password since its more secure and Primavera does not have an appropriate authentication system.

Core views - Information architecture

- Product view
- Customer view
- Orders View
- Catalog View
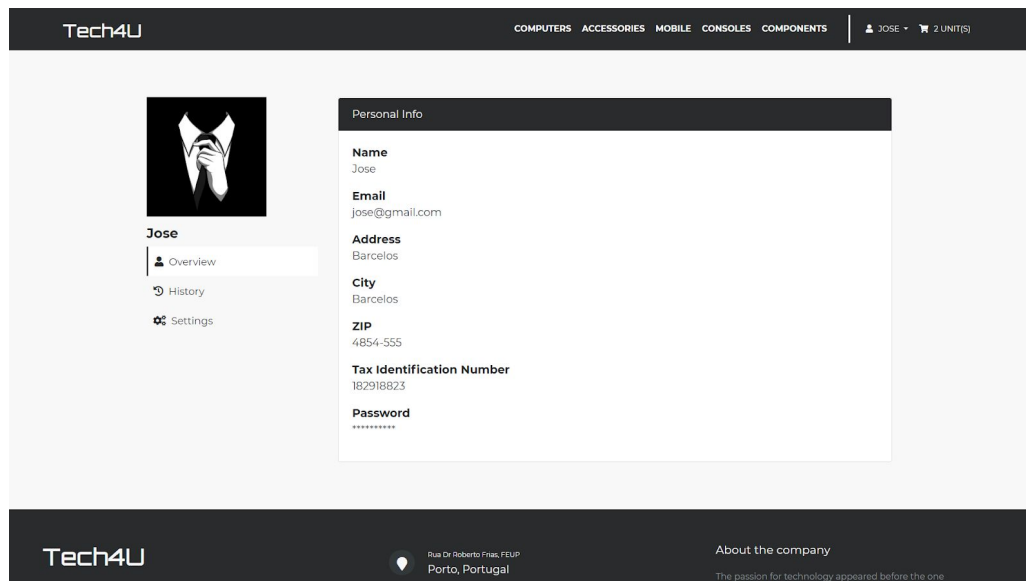- Shopping Cart View
- Register View
- Administrator View
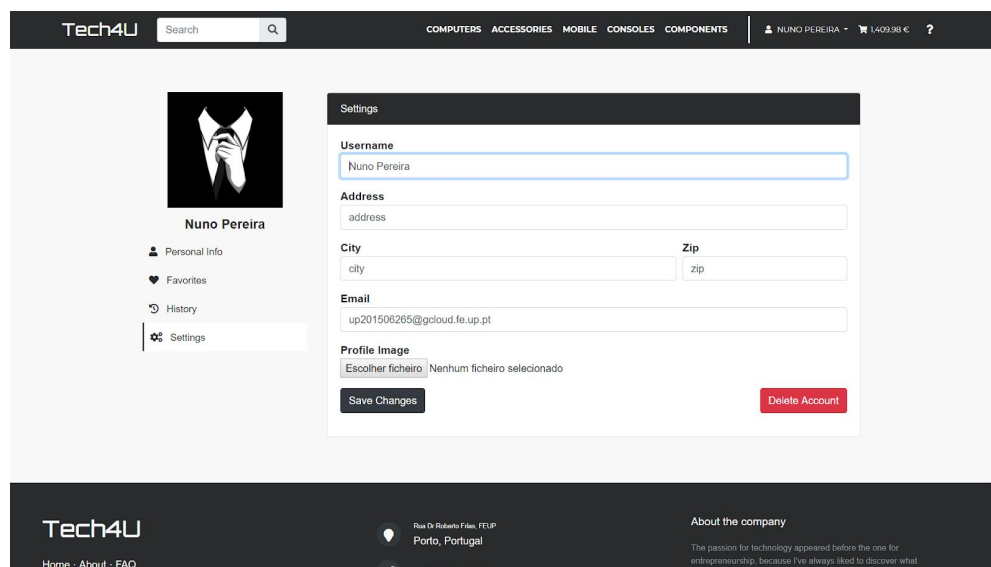
Product view

Elements of the core

- Product overview (name, price, description, score)
- Button 01: Add product to cart

Customer view

Elements of the core:

- Customer overview
- Button 02 - Shopping history
- Button 03 - Edit profile information

Elements of the core:

- Form 01: Edit profile form
- Button 01: Escolher ficheiro
- Button 02: Save changes
- Button 03: Delete account
- Button 04 - View personal info
- Button 05 - Shopping history

Shopping cart

Elements of the core

- Table 01: List of products in the shopping cart
- Button 01: Checkout
- Button 02: Delete product from shopping cart
- Button 03: Increase quantity
- Button 04: Decrease quantity

Elements of the core

- Form 01: Expedition form
- Button 01: Next step
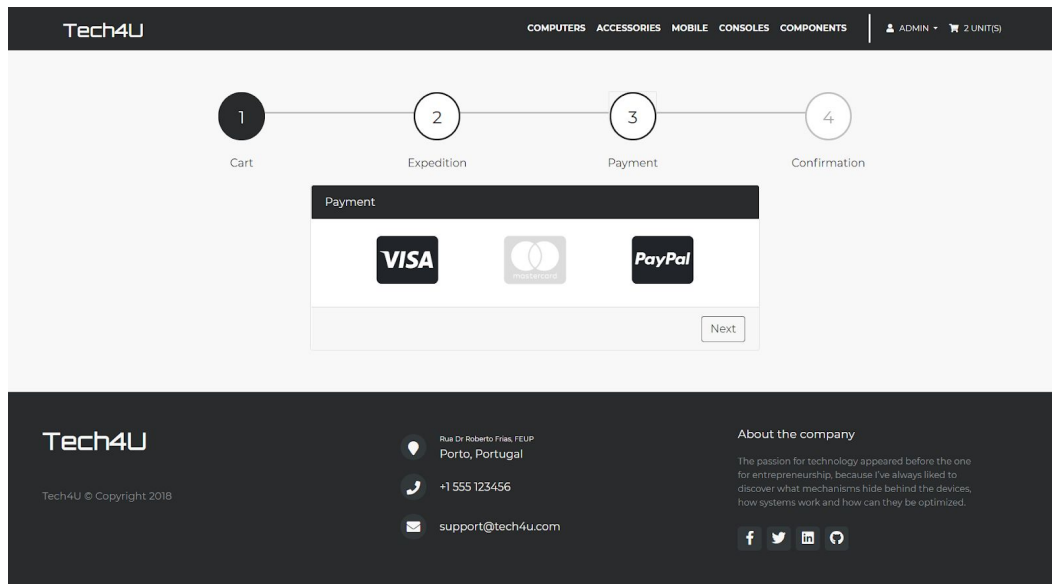
Elements of the core

- Table 01: Method of payment
- Button 01: Next step
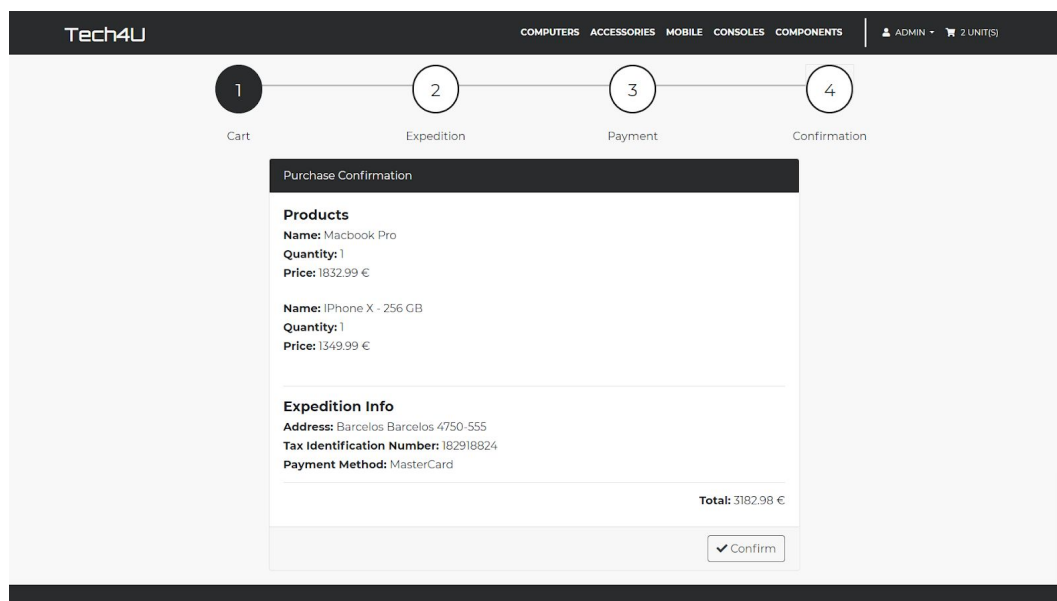
Elements of the core

- Table 01: Confirmation details
- Button 01: Confirm button (Finish order)

Orders View
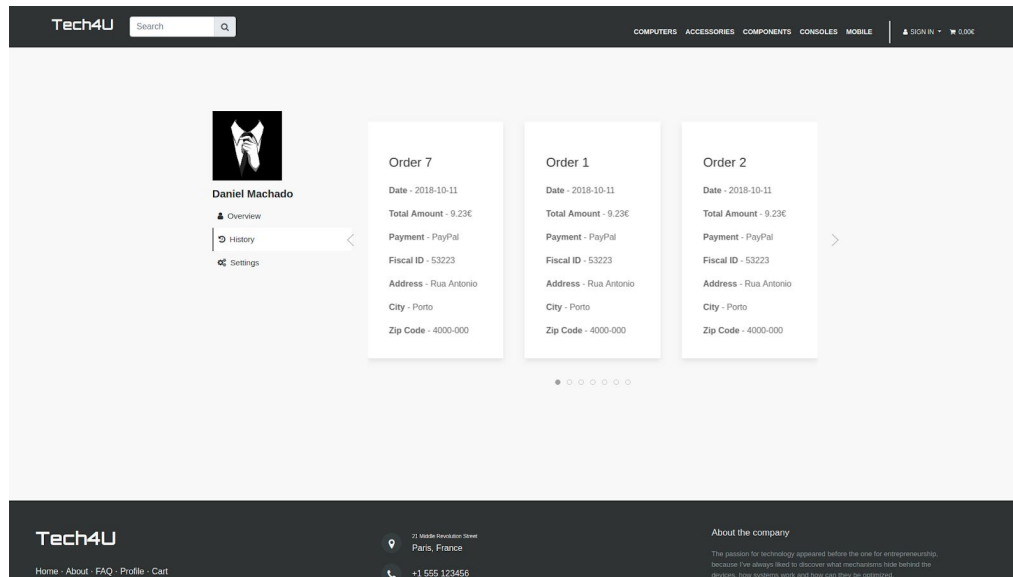
Elements of the Core
- List 01: Orders List
- Table 01: Orders Info

Registration View

Elements of the Core

- Form 01: Sign up form
- Checkbox 01: Accept Terms and Conditions

- Button 01: Sign up button

Catalog View
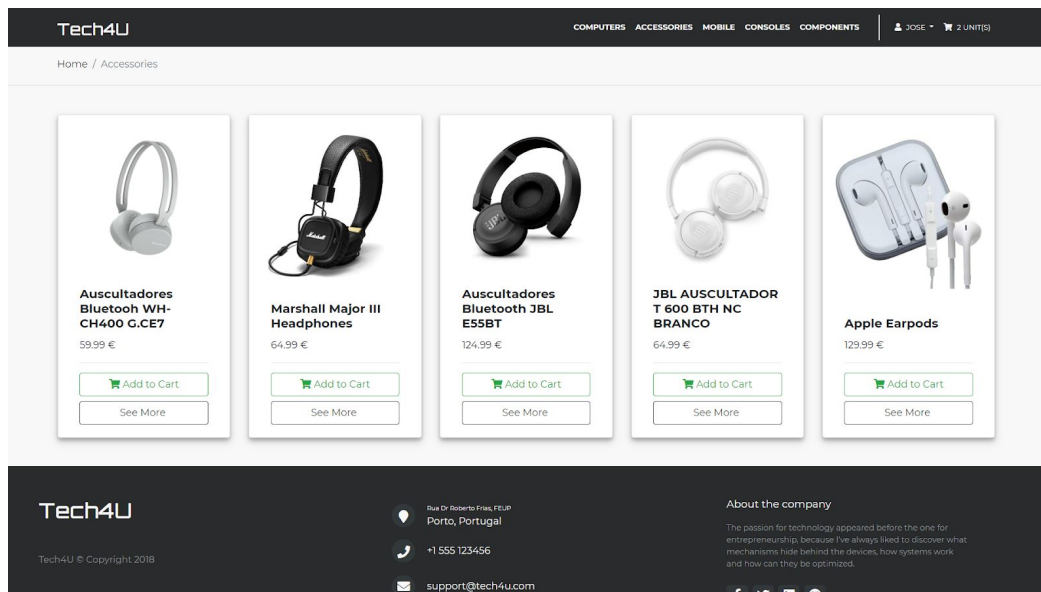
Elements of the core

- Button 01: Add product to cart
- Button 02: See product page
- Image 01: Product image

Administrator View
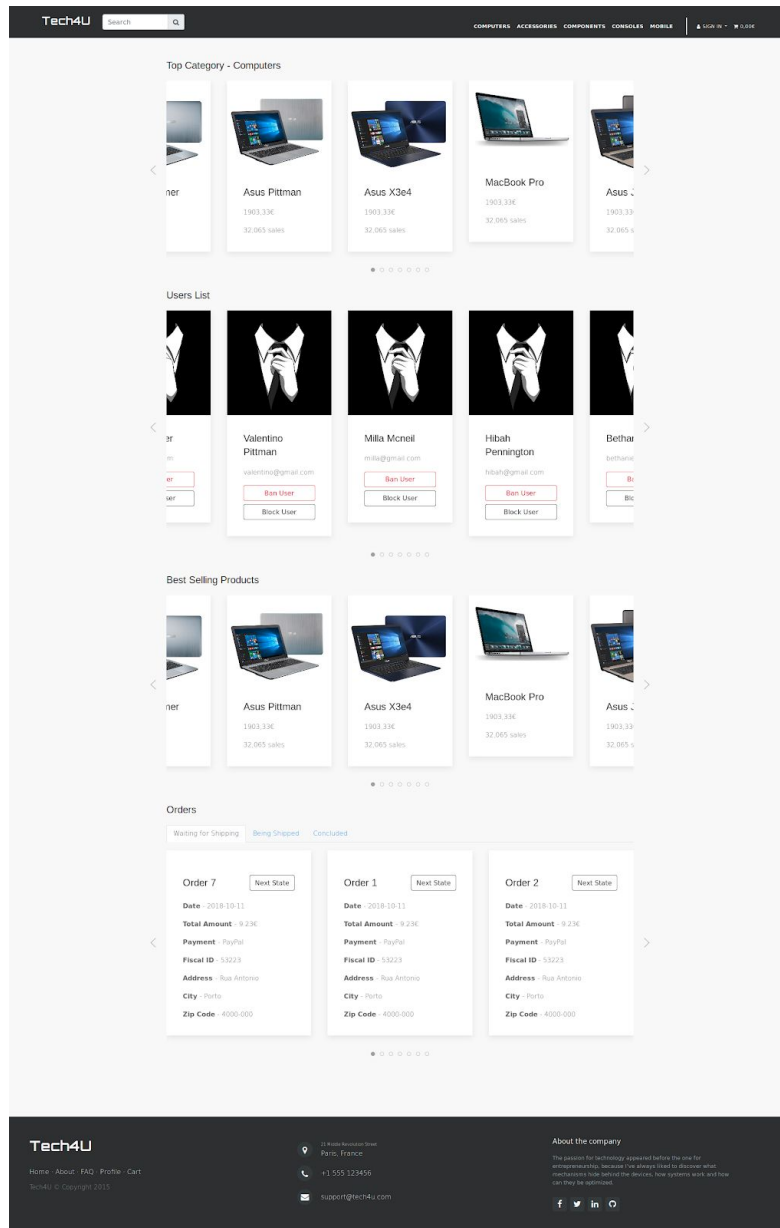
Elements of the core

- List 01: Top Category
- List 02: Users List
- Button 01: Ban user
- Button 02: Block user
- List 03: Best Selling Products
- List 04: Orders List
- Button 03: Next Step

Other Features

- Implemented a new local database for a better management of users and permissions;

Interoperability with Primavera WebApi

Webservice ID	Webservice Description	Related Core Views(s)	Input	Output
get_item	Returns a product with the specified ID.	Product View	**Route: GET** <IP>/WebApi/Administrador/Consulta **Body:** SELECT a.Artigo, a.Descricao, am.PVP1, m.Descricao, a.Observacoes, a.Peso, f.Descricao FROM Artigo a INNER JOIN Marcas m ON m.Marca=a.Marca INNER JOIN Familias f ON f.Familia=a.Familia INNER JOIN ArtigoMoeda as am ON a.Artigo = am.Artigo WHERE a.Artigo=<productID>	
| get_categories | Returns all categories of products | All Views | **Route: GET** <IP>/WebApi/Administrador/Consulta<br><br>**Body:** SELECT Familia, Descricao FROM Familias | ```
{
    "DataSet": {
        "Table": [
            {
                "Familia": "F001",
                "Descricao": "Computers"
            },
            {
                "Familia": "F002",
                "Descricao": "Mobile"
            },
            {
                "Familia": "F003",
                "Descricao": "Consoles"
            }
        ]
    },
    "Query":
"System.Data.SqlClient.SqlCommand"
}
``` |
| get_category | Returns all products form a specified category, alongside with some relevant information about the products | Catalog View | **Route: GET** <IP>/WebApi/Administrador/Consulta

Body: SELECT a.Artigo, a.Descricao, am.PVP1 FROM Artigo as a INNER JOIN Familias ON a.Familia = Familias.Familia INNER JOIN ArtigoMoeda as am ON a.Artigo = am.Artigo WHERE Familias.Descricao =<categoryID> | |
| create_client | When a user concludes his | Registration View | **Route: POST** <IP>/WebApi/Base/Clientes/Atualiza

Body: | N/A |

	registration a new client must be created with the data provided		{
			    Cliente: <ID>,
			    Nome: <NAME>,
			    Morada: <ADDRESS>,
			    Localidade: <CITY>,
			    CodigoPostal:<ZIP>,
			    LocalidadeCodigoPostal: <CITY>,
			    NumContribuinte: <NIF>,
			    CondPag: 2,
			    Moeda: "EUR",
			    Pais: "PT"
			}	

get_client	Returns the client with the specified ID	Customer View	**Route: GET** <IP>/WebApi/Administrador/Consulta **Body:** SELECT Fac_Mor, Fac_Local, Fac_Cp, NumContrib FROM Clientes WHERE Clientes.Cliente=<userID>	

update_client	Allows the user to change his information in the system	Customer View	**Route: POST**
			<IP>/WebApi/Base/Clientes/Atualiza
			
			Body:
			{
			
			    CodigoTabLog: "Cliente",
			    ChaveLog: "Cliente",
			    Cliente: <ID>,
			    Nome: <NAME>,
			    Morada: <ADDRESS>,
			    Localidade: <CITY>,
			    CodigoPostal:<ZIP>,
			    LocalidadeCodigoPostal: <CITY>,
			    NumContribuinte: <NIF>,
			    CondPag: 2,
			    Moeda: "EUR",
			    Pais: "PT",
			    EmModoEdicao: true
			}
				N/A

get_client_orders	Returns all order the client has placed alongside with the status of the order.	Orders View	**Route: GET** <IP>/WebApi/Administrador/Consulta **Body:** SELECT CONVERT(VARCHAR(10),cd.Data,103), cd.DataCarga, cd.NomeFac, cd.Id, cd.TotalMerc, cd.TotalIva, cd.TotalDocumento, cd.ModoPag, cd.NumContribuinte, cd.MoradaEntrega, cd.LocalidadeEntrega, cd.CodPostalEntrega, cds.Estado FROM CabecDoc cd INNER JOIN CabecDocStatus cds ON cd.id = cds.IdCabecDoc WHERE cd.Entidade=<userID> AND cd.TipoDoc="ECL"	
get_order_products	Return all products, their respective	Orders View	**Route: GET** <IP>/WebApi/Administrador/Consulta **Body:** SELECT cd.TotalDocumento, cd.NumContribuinte, cd.TotalIva, cd.Morada,	

	quantities, unit price and total price from a specified order		cd.Localidade, cd.CodPostal, ld.TaxaIva, ld.descricao, ld.quantidade, ld.PrecUnit, ld.PrecoLiquido, ld.TaxaIva, ld.TotalIva FROM LinhasDoc as ld INNER JOIN CabecDoc as cd ON ld.IdCabecDoc = cd.Id WHERE ld.IdCabecDoc =<orderID>	

create_order	When a customer finishes a purchase a document with the specified fields must be created to save his order in the system	Shopping Cart View	**Route: POST** <IP>/WebApi/Vendas/Docs/CreateDocument/
			
			Body:
			{
			  "Linhas": [
			  {
			   "Artigo": <Item>,
			   "Quantidade": <Quantity>
			  }],
			  "Tipodoc": "ECL",
			  "Entidade": <ID>,
			  "TipoEntidade": "C",
			  "ModoPag": <PAYMENT>,
			  "MoradaEntrega": <ADDRESS>,
			  "LocalidadeEntrega": <CITY>,
			  "CodPostalEntrega": <ZIP>,
			  "DataVenc": <DATE>,
			  }
				N/A

get_best_sellers	Return the top X of the best selling products, X must be specified by the Administrator	Administrator View	**Route: GET** <IP>/WebApi/Administrador/Consulta **Body:** SELECT TOP(4) Artigo, Descricao, PrecUnit, SUM(Quantidade) AS TotalQuantity FROM LinhasDoc GROUP BY Artigo, Descricao, PrecUnit ORDER BY SUM(Quantidade) DESC	

get_top_category	Returns the most purchased category	Administrator View	**Route: GET** <IP>/WebApi/Administrador/Consulta
			
			Body:
			SELECT TOP(1) f.familia, f.descricao FROM Familias as f INNER JOIN Artigo as a ON f.familia=a.familia INNER JOIN LinhasDoc as ld ON ld.artigo=a.artigo GROUP BY f.familia,f.descricao ORDER BY SUM(ld.quantidade)
				{
				  "DataSet": {
				   "Table": [
				    {
				     "familia": "F001!
				     "descricao": "Computers"
				    }
				   ]
				  },
				  "Query":
				"System.Data.SqlClient.SqlCommand"
				}

get_transform_orders	Returns all the transformed orders	Administrator View	**Route: GET** <IP>/WebApi/Administrador/Consulta **Body:** SELECT CONVERT(VARCHAR(10),cd.Data,103), cd.TotalMerc, cd.TotalIva, cd.TotalDocumento, cd.ModoPag, cd.NumContribuinte, cd.MoradaEntrega, cd.LocalidadeEntrega, cd.CodPostalEntrega, cds.Estado FROM	

			CabecDoc cd INNER JOIN CabecDocStatus cds ON cd.id = cds.IdCabecDoc WHERE cds.Estado ="T"	
get_peding _orders	Returns all pending orders	Administ rator View	**Route: GET** <IP>/WebApi/Administrador/Consulta **Body:** SELECT CONVERT(VARCHAR(10),cd.Data,103), cd.NumDoc, cd.Entidade, cd.TotalMerc, cd.TotalIva, cd.TotalDocumento, cd.NumContribuinte, cd.MoradaEntrega, cd.LocalidadeEntrega, cd.CodPostalEntrega, cds.Estado FROM CabecDoc cd INNER JOIN CabecDocStatus cds ON cd.id = cds.IdCabecDoc WHERE cds.Estado ="P" AND cd.TipoDoc="ECL"	
manage_pr oducts	Return a list with all the products and their respectiv e stock	Administ rator View	**Route: GET** <IP>/WebApi/Administrador/Consulta **Body:** SELECT a.Artigo, a.Descricao,aa.Armazem, SUM(aa.StkActual) FROM Artigo a INNER JOIN V_INV_ArtigoArmazem aa ON aa.Artigo = a.Artigo GROUP BY a.Artigo,a.Descricao,aa.Armazem	

create_sto ck_order	Place a new order to update the stock of a product	Administ rator View	**Route: GET**
			<IP>/WebApi/Compras/Docs/CreateDocument
			
			Body:
			{
			  "Linhas": [{"Artigo": <ID>, "Quantidade":<QUANTITY>}],
			  "Tipodoc": "ECF",
			  "Entidade": "F001",
			  "TipoEntidade": "F",
			  "CondPag": 2
			}
				N/A

create_ite m	Creates an item with the specified fields	Administ rator View	**Route: POST**
			<IP>/Base/Artigos/Actualiza/
			
			Body:
			{
			  "Artigo": <ID>,
			  Descricao: <NAME>,
			  Caracteristicas: "",
			  Observacoes: <DESCRIPTION>,
			  CodBarras: "",
			  UnidadeBase: "UN",
			  UnidadeVenda: "UN",
			  UnidadeCompra: "UN",
			  UnidadeEntrada: "UN",
			  UnidadeSaida: "UN",
			  Marca: <BRAND>,
			  Familia: <CATEGORY>,
			  Peso:<SCORE>,
			  IVA: "23",
			  MovStock: "S"
			}
				N/A

create_ite m_price	When a new item is create another route must be called in order to associat e a price to that item	Administ rator View	**Route: POST**
			<IP>/Base/ArtigosPreco/Actualiza/
			
			Body:
			{
			"Artigo": <ID>,
			  "Moeda": "EUR",,
			  "Unidade": <UN>,
			  "Descricao": "",
			  "PVP1": <PRICE>
			}
				N/A

transform_ doc	Docume nts created when a user finishes an order are of type "ECL" these must be transfor m in docume nts of type FA after the payment is validated	Administ rator View	**Route: POST**
			<IP>/Vendas/Docs/TransformDocument/ECL/A/< docID>/000/true
			
			Body:
			{
			  "Tipodoc": "FA",
			  "Serie": "A",
			  "Entidade": <CLIENT>,
			  "TipoEntidade": "C",
			}
				true

NOTE: A service to remove clients, products or any other has not been implemented since keeping this information in the database, even though they are not relevant to the app anymore, can be useful for statistical purposes.

Paths

System Architecture

Project Specifications vs Delivered Project

As the project evolved we realized that some of the features we have specified in previous deliveries would require little or no interaction with ERP Primavera. Features like reviews, wishlist and item score although relevant in an online shop did not seem to add any interoperability with ERP Primavera. As such we decided to invest in more specific and elaborated "Administrator Dashboard" instead.

We made available to the Administrator of our site different sets of information about sales and products, such as "Top Category" and "Best Selling Products", also the Administrator is able to easily manage users, products and orders.

Besides that, in our opinion online stores are all about users experience, so we tried to make our website as user friendly as possible. Making it 100% responsive, very interactive so the user is always aware of his interaction with the app and (again in our opinion) easy on the eye. Also to improve the user experience, and because there was no appropriate way to use Primavera to this meaning, the shopping cart is stored solely in cookies until the moment the user places an order this allows him to add items to the cart even without being authenticated and his choices to endure even if the browser is closed.

Lessons Learned