

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

Escola de Ciências e Tecnologia

Solução de retalho para a ótica em LS Business Central

Relatório de Estágio de Mestrado em Engenharia Informática

Micaela Dara Silva do Cima

Orientador:

Professor Doutor Luís Filipe Leite Barbosa

Coorientador:

Engenheiro Jorge Luís Rocha dos Santos



Vila Real, 2020

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

Escola de Ciências e Tecnologia

Solução de retalho para a ótica em LS Business Central

Relatório de Estágio de Mestrado em Engenharia Informática

Micaela Dara Silva do Cima

Orientador:

Professor Doutor Luís Filipe Leite Barbosa

Coorientador:

Engenheiro Jorge Luís Rocha dos Santos

Vila Real, 2020

Orientador:

Professor Doutor Luís Filipe Leite Barbosa

Coorientador:

Engenheiro Jorge Luís Rocha dos Santos

Composição do Júri:

Presidente:

Doutor Hugo Alexandre Paredes Guedes da Silva

Vogais:

Doutor João Paulo Pereira de Sousa

Doutor Arsénio Monteiro dos Reis

Doutor Luís Filipe Leite Barbosa

Resumo

No ambiente empresarial, é necessário a presença de um *software* que ajude a melhorar o funcionamento interno com funcionalidades compatíveis com o trabalho pois é importante para melhorar o desempenho dos seus funcionários. No ambiente empresarial, é necessário o controlo de diversos processos com o objetivo de tornar um produto ou serviço mais atrativo para o cliente. Para isso, é necessário a presença de um *software* que ajude a um maior controlo e compressão de todos os processos envolvidos, para facilitar a gestão e o desenvolvimento de produtos/ serviços com maior qualidade.

Assim se houver um *software* que consiga fazer a gestão de todas as funcionalidades da loja, esta aumenta a produtividade da empresa.

O software de LS Retail, possui uma gama elevada de funcionalidades com o uso do Microsoft Dynamics Business Central 365 (BC), este permite fazer a gestão de produtos, stock, emitir faturas, fazer gestão de armazém entre muitas outras funcionalidades. Contudo este pode ser sempre adaptado a cada necessidade da empresa para melhorar os processos internos da mesma.

O LS Otical, é um *add-on* que foi adicionado às funcionalidades do BC, para permitir que se consiga fazer encomendas específicas de óculos, para que os funcionários possam escolher cada um dos componentes dos óculos, a sua graduação e outras especificações.

O mundo das aplicações móveis está em crescimento e as compras feitas a partir do sofá de casa estão cada vez mais na moda, pois assim as pessoas não precisam de sair de casa para fazer as suas compras e estas podem também ser recebidas no conforto de sua casa ou levantadas em loja.

Com o complemento da funcionalidade de adição de encomendas de óculos na aplicação do LS Retail, permitiu que os clientes possam efetuar as encomendas dos seus óculos a partir do conforto de sua casa.

Neste relatório de estágio são apresentados conceitos do LS Retail, quais os seus benefícios, algumas aplicabilidades e quais os riscos que este possui. É feita também uma apresentação do LS Otical bem como as suas funcionalidades.

Por fim é feita uma apresentação de toda a implementação das diferentes partes deste trabalho e de todos os problemas que surgiram, bem como, as soluções encontradas para a sua resolução.

Palavras Chave: LS Retail, LS Otical, Microsoft Dynamics Business Central, AL, Aplicações Móveis;

Abstract

In the business environment, it is necessary to have software that helps to improve the internal functioning with functionalities compatible with the work as it is important to improve the work of your employees.

If there is software that manages all the store's features, it increases the company's productivity.

The LS Retail software has a wide range of features using Microsoft Dynamics Business Central (BC), which allows you to manage products, stock, issue invoices, manage warehouse and many other features. However, it can always be adapted to each company's needs to improve its internal processes.

LS Otical, is an add-on that has been added to the BC functionalities, to allow you to be able to place specific orders for glasses, so that employees can choose each component of the glasses, their graduation and other specifications.

The world of mobile applications is growing and purchases made from the sofa at home are more and more fashionable, so that people do not need to leave home to do their shopping and these can also be received in the comfort of their home or raised in store.

With the addition of the functionality for adding glasses orders to the LS Retail application, it has enabled customers to place orders for their glasses from the comfort of their home.

In this internship report LS Retail concepts are presented, what are its benefits, some applicability and what risks does it have. There is also a presentation of LS Otical as well as its functionalities.

Finally, a presentation is made of the entire implementation of the different parts of this work and of all the problems that existed, as well as the solutions for its resolution.

Keywords: LS Retail, LS Otical, Microsoft Dynamics Business Central, AL, Mobile Applications;

Agradecimentos

Gostaria de manifestar o meu agradecimento a todas estas pessoas que me apoiaram neste percurso:

Obrigado à Universidade de Trás-os-Montes e Alto Douro (UTAD) por todo o apoio institucional, sob a forma de meios e serviços, que me disponibilizou, bem como aos seus docentes, que tiveram um papel na minha formação.

Quero deixar um agradecimento especial ao meu orientador, o Professor Doutor Luís Barbosa, por todo o apoio teórico que me deu, sem o qual não teria sido possível escrever a parte teórica do trabalho.

Quero deixar uma palavra de agradecimento à ARQUICONSULT e a Arquiconsultnearshore, pela oportunidade que me deram de desenvolver o meu projeto de estágio em um ambiente de empresa, ter-me dado a oportunidade de participar neste projeto de estágio desafiante.

Gostaria de deixar um agradecimento especial ao Chefe de equipa André Botelho por toda a ajuda dada em momentos de maior dificuldade no desenvolvimento da parte prática e ao Valdemar Faria pela oportunidade de fazer este projeto que me fez ganhar muito conhecimento ao longo de todo o seu desenvolvimento.

Quero agradecer também ao meu coorientador pelo apoio no desenvolvimento deste projeto.

Gostaria também de agradecer ao meu colega de trabalho Rui Pires e a toda a equipa da ArquiconsultNearshore por todo o apoio dado ao longo do desenvolvimento deste trabalho.

Pretendo deixar o maior agradecimento possível aos meus pais, ao meu irmão, pois, sem um bom ambiente familiar e sem uma amizade como a que temos, eu não seria a pessoa que sou hoje. Um bem-haja, à minha restante família que sempre demonstrou uma grande estima por mim e sempre desejou o meu sucesso.

Por fim queria também agradecer a todos os meus amigos que estiveram sempre comigo ao longo de todo o percurso académico.

A todos,
MUITO OBRIGADO Micaela Cima

Aos meus pais, que sempre estiveram do meu
lado e acreditaram em mim.

Índice

1 - Introdução	1
1.1 - Arquiconsult	2
1.2 - LS Retail	2
1.3 - Motivação e Objetivos	5
1.4 - Organização do Documento	6
1.5 - O Estágio	6
2 - Revisão do Estado da Arte	9
2.1 - Vantagens e Desvantagens de um Sistema ERP	10
2.2 - Tendências de ERP	12
2.3 - Descrição Geral do Produto – Microsoft Dynamics 365 Business Central	13
2.3.1 - Visão Tecnológica	15
2.4 - LS RETAIL – Ótica	20
2.4.1 - Visão Geral das Óticas	21
2.5 - Desafios na Área do Retalho	21
2.6 - SF6	23
2.7 - Testes de Software	25
2.7.1 - Como Realizar um Teste de Software	27
2.7.2 - Tipos de Testes de Software	28
3 - Protótipo Implementado	31
3.1.1 - Arquitetura do Sistema	31
3.1.2 - Análise de Requisitos da Migração	31
3.1.3 – Diagrama de Casos de Uso da Migração	32
3.2 - Implementação da Extensão	35
3.2.1. Migração de BC para AL	35
3.3 – Software usado na Migração	35
3.3.1 - Structured Query Language	36
3.3.2 - POWERSHELL	38
3.3.3 - Visual Studio Code	39
3.3.4 – Microsoft Dynamics 365 Business Central	46
3.4 - Implementação do SF6	49
3.5 - Criação da Aplicação Móvel	51
3.5.1 - Análise de Requisitos da Aplicação LS Retail Loyalty	53
3.5.2 - Diagrama de Casos de Uso	54
3.6 - Softwares Usados na Aplicação LS Retail Loyalty	55
3.6.1 - Visual Studio 2019	55

3.6.2-LS OMNI.....	55
3.6.3-WCF	56
3.7- Implementação da Aplicação	57
3.8 - Dificuldades de Implementação	64
3.9-Testes de Software Realizados no Desenvolvimento das Duas Componentes	65
4 - Conclusões	73

Lista de Figuras

Figura 1: Empresas que utilizam o LS Retail. Adaptado de LSRetail, 2020.....	3
Figura 2: Arquitetura de sistema do LS Retail (Planet, 2020).....	3
Figura 3: Arquitetura do sistema LS Retail. Adaptado de LSRetail, 2020.	4
Figura 4: Desafios do LS Retail. adaptado de LS Retail, 2020.....	5
Figura 5: Estrutura do Microsoft Dynamics Nav, de (Brodie, 2017)	10
Figura 6: Exemplos de ERP, adaptado de (PRO, 2020)	12
Figura 7: Centro perfil, Microsoft Dynamics Nav.	13
Figura 8: Cliente web	14
Figura 9: Cliente Tablet.....	14
Figura 10: Estrutura do Microsoft Dynamics NAV, de (Brodie, 2017)	16
Figura 11: Cloud para Microsoft.	17
Figura 12: Ambiente de desenvolvimento NAV.	18
Figura 13: Ambiente de cliente NAV.	19
Figura 14 :Imagem ilustrativa de ferramentas disponibilizadas pelo NAV.	20
Figura 15: Exemplo da realização de testes, adaptado de (Naik, 2011).....	27
Figura 16: Exemplo de testes de software, adaptado de (Caetano).	28
Figura 17: Arquitetura do sistema. (Retail, 2021)	31
Figura 18: Criação de fields.	41
Figura 19: Definição de chaves primárias.	41
Figura 20: Exemplo de trigger.	42
Figura 21: Exemplo de procedure.	42
Figura 22: Definição de variáveis globais.	42
Figura 23: Exemplo de erro em nomes.	42
Figura 24: Exemplo de correção de erro.	42
Figura 25: Exemplo de criação de pages.	43
Figura 26: Exemplo de inserção de campos em pages.	43
Figura 27: Definição de part pages.	44
Figura 28: Exemplo de criação de actions.	44
Figura 29: Exemplo de variáveis dotnet	45
Figura 30: Exemplo de Formato do import	46
Figura 31: Definição de campos no xmlport	46
Figura 32: Exemplo de um documento xmlPort em AL.	49
Figura 33: Exemplo de um documento no formato SF6.	50
Figura 34: Exemplo da importação do formato SF6, por loja.....	50
Figura 35: Exemplo de lojas existentes.....	51
Figura 36: Exemplo de algumas categorias	52
Figura 37: Exemplo de um ecrã da aplicação	52
Figura 38: Apresentação do LS OMNI (Retail L. , 2021)	56
Figura 39: Menu principal com link para o ecrã da ótica	58
Figura 40: Ecrã inicial	59
Figura 41: Ecrã com informação dos óculos	59
Figura 42: Ecrã para escolher a rarefação das lentes da esquerda	60
Figura 43: Ecrã para escolher a rarefação das lentes da direita.....	60
Figura 44: Ecrã para selecionar as hastes e as lentes e para adicionar os produtos ao carrinho.....	61
Figura 45: Seleção dos itens por frame.....	62
Figura 46: Categoria das Frames.....	62
Figura 47: Sub-Categorias das Frames	62

Figura 48: Adicionar um item há encomenda	63
Figura 49: Criação de clientes na migração	66
Figura 50: Escolha do índice de rarefação das lentes.....	66
Figura 51: Escolha dos produtos das lentes e das hastes.....	67
Figura 52: Alteração do estado das glass sales orders	68
Figura 53: Exemplo da fatura do cliente.....	68
Figura 54: Ecrã para fazer ping na app.....	69
Figura 55: Resposta ao Ping.....	69
Figura 56: Codeunit do OmniWrapper.....	70
Figura 57: Página para fazer o test connection	70
Figura 58: Resposta ao test connection	71

Lista de Tabelas

Tabela 1 - Abreviações usadas pelos fabricantes de lentes.....	24
Tabela 2 - Comandos SQL	36
Tabela 3 - Alguns comandos PowerShell.....	39

Acrónimos

AL	-	<i>Application Language</i>
BC	-	<i>Microsoft Dynamics 365 Business Central</i>
BC365	-	<i>Microsoft Dynamics 365 Business Central</i>
C/AL	-	<i>Client/server Application Language</i>
CMM	-	<i>Capability Maturity Model</i>
CRM	-	<i>Customer relationship management</i>
DLL	-	<i>Dynamic-link library</i>
DO	-	<i>Dynamics Optical</i>
Dynamics AX	-	<i>Dynamics 365 for Finance and Operations</i>
Dynamics NAV	-	<i>Dynamics Navision</i>
ERP	-	<i>Enterprise Resource Planning</i>
IDE	-	<i>Integrated Development Environment</i>
ISO	-	<i>International Organization for Standardization</i>
IIS	-	<i>Internet Information Services</i>
LS	-	<i>LS RETAIL</i>
MRP	-	<i>Manufacturing Requirements Planning</i>
NAV	-	<i>Navision</i>
OCR	-	<i>Optical Character Recognition</i>
ODATA	-	<i>Open Data Protocol</i>
POS	-	<i>Point of Sale/Point of Service</i>
RSS	-	<i>Really Simple Syndication</i>
SF6	-	<i>B2BOptic LensCatalog</i>
SOA	-	<i>Service-Oriented Architecture</i>
SOAP	-	<i>Simple Object Access Protocol</i>
VB	-	<i>Visual Basic</i>
WCF	-	<i>Windows Communication Foundation</i>
XML	-	<i>Extensible Markup Language</i>

1 - Introdução

Face ao crescimento abrupto da tecnologia nos últimos tempos, principalmente na área do retalho, a utilização de sistemas Enterprise Resource Planning – ERP veio permitir a ajuda na gestão de processos dentro de uma empresa. Os sistemas de informação estão em contínua evolução. Em pouco tempo houve uma evolução que constituiu no aparecimento do MRP – Manufacturing Requirements Planning, uma ferramenta que foi criada para ajudar as empresas na gestão dos custos ao longo do tempo, ajudando a otimizar alguns processos de negócio atingindo posteriormente o ERP (Stair, 1998). A utilização do ERP tem como foco facilitar o fluxo de informações de uma organização permitindo assim uma organização de informação em diversos níveis e, oferecendo em tempo real indicadores de performance que ajudam nessa gestão. Muitas empresas optam pelos serviços ERP devido a vários motivos, sendo que o principal objetivo é unificar os processos, combatendo a incompatibilidade de sistemas. O setor das óticas está em fase de crescimento e de grande demanda o que leva à necessidade da integração de um *software* dotado de capacidades para permitir gerir toda a sua lógica de negócio de uma forma simples. É assim possível tornar mais acessível ao cliente fazer encomendas aos fornecedores; efetuar pagamentos; receber notificações de entrega do material, no final da montagem do material; avisar os clientes que podem levantar o seu produto; finalizar compra e passando fatura ao mesmo (Padilha & Marins, 2005).

Os dispositivos móveis estão cada vez mais presentes nas nossas vidas através dos computadores, *smartphones*, *wearables*, entre outros, tendo estes mais funcionalidades e poder de processamento que nos permitem facilitar o nosso dia-a-dia. O desenvolvimento de aplicações para estes dispositivos tem aumentado nas mais diversas áreas que vão desde o turismo, medicina, ensino, comércio eletrónico, entre outras (Miguel Mazedo, 2020), o que tem sido comprovado pela existência de aplicações de todo o género, entre as quais aplicações relacionadas com LS Retail/Oticas (Ferreirinha, 2017). A forma *standard* é o desenvolvimento de aplicações nativas para cada uma das plataformas, de forma independente, usando as tecnologias que a própria marca disponibiliza (Java usando Android Studio para Android, Objective-C ou Swift usando o Xcode para iOS, e C# usando Visual Studio para desenvolver para as plataformas Windows). (Miranda, 2017)

1.1 - Arquiconsult

O meu estágio decorreu na Arquiconsult, é uma empresa de consultoria de sistemas de informação, assente em tecnologias Microsoft, com o título de “Gold Certified Partner [Microsoft Gold Enterprise Resource Planning, Microsoft Gold Application Development, Microsoft Gold Cloud Platform]”.

Atualmente conta com um grupo de mais de 140 consultores, portadores de um elevado conhecimento em integrações com outros sistemas e, por isso, são frequentemente referidos a clientes internacionais para as suas implementações.

A Arquiconsult tem já uma vasta experiência na implementação de complexas Soluções de Negócio Microsoft, nomeadamente em Microsoft Dynamics 365 Business Central (. Dynamics NAV) e Dynamics 365 for Finance and Operations (Dynamics AX).

1.2 - LS Retail

LS Retail fornece soluções simples para a gestão empresarial. As soluções são altamente configuráveis e ajudam retalhistas e hoteleiros, em todo o mundo a otimizar as suas práticas de negócio, aumentando as receitas e a satisfazer novos e antigos clientes, fazendo que estes se mantenham fidelizados com a empresa. Sendo mais fáceis de usar, mais simples e mais rápidas estas soluções permitem que a empresa consiga fazer uma gestão pormenorizada da sua rede de lojas. (Asserbiz, 2016)

Desde os primórdios, as empresas têm como principal objetivo a ambição de crescimento, encarando novas maneiras de negócio de uma forma flexível e cooperativa. Para tal deverão possuir processos, mentalidades e ferramentas que facilitem a sua integração na economia que as rodeia. É neste ponto que se enquadra as tecnologias Microsoft Dynamics.

Existem diversas empresas a usar Ls Retail, tais como, CAFÉ la MODE, Blue Mountain, Army Navy Burger and Burrito, Gallo Clothing, Yves-Rocher, Scandlines, Juventus F.C. entre outras, Figura 1.



Figura 1: Empresas que utilizam o LS Retail. Adaptado de LSRetail, 2020.

No topo, há a camada de negócios (Head Office), que é o Microsoft Dynamics Business Central 365, que inclui a funcionalidade padrão do ERP. O Backoffice aprimora ou adiciona funcionalidades necessárias aos negócios de LS Retail, como podemos ver na figura 2 e na 3, mas também possui desafios como podemos ver na figura 4.

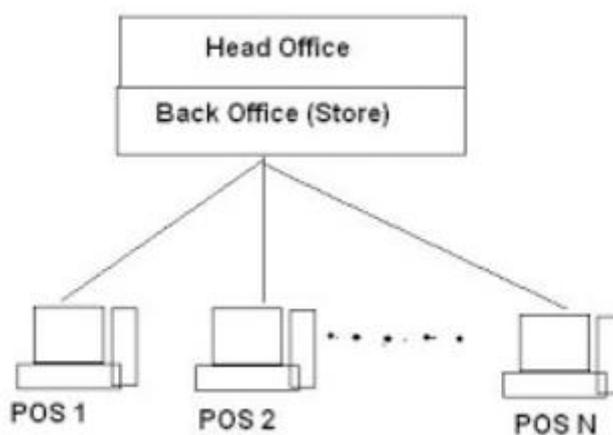


Figura 2: Arquitetura de sistema do LS Retail (Planet, 2020)

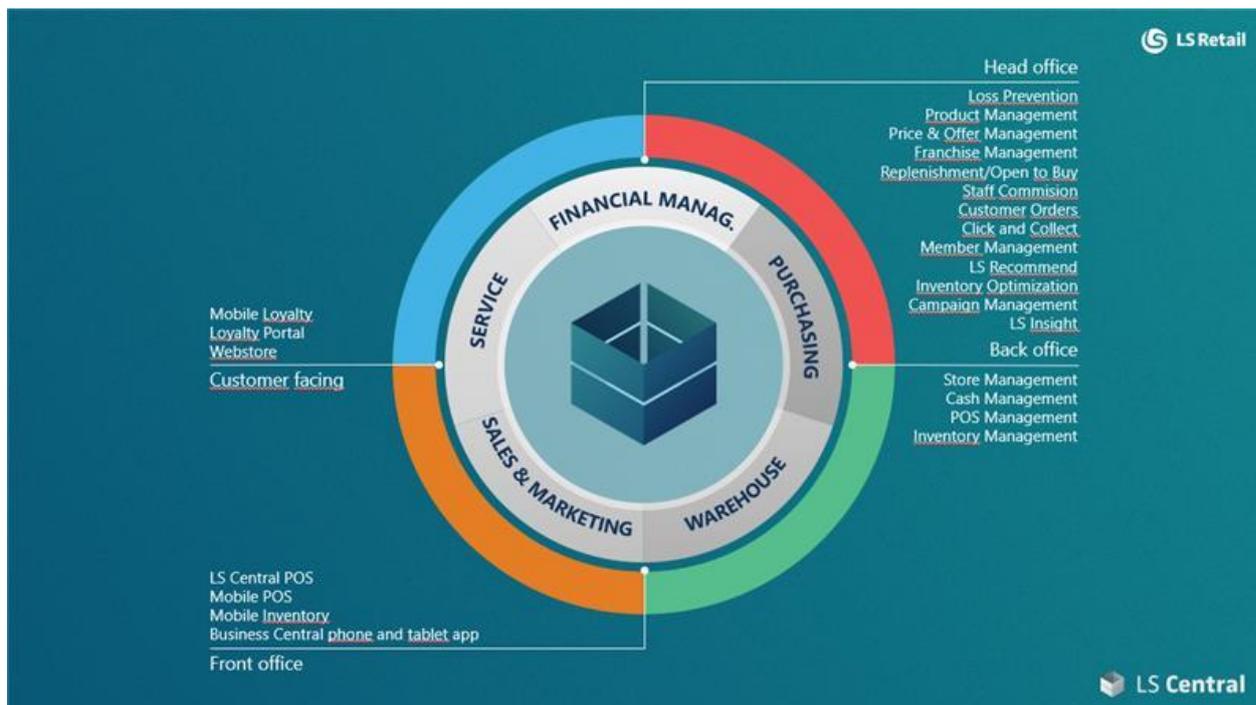


Figura 3: Arquitetura do sistema LS Retail. Adaptado de LSRetail, 2020.

Com isto pode-se gerir centralmente os importantes complementos do LS Retail, além da Central de Negócios padrão.

Se separarmos a solução por locais de trabalho, primeiro teremos a funcionalidade da matriz, projetada para atender às necessidades e requisitos das operações centrais.

Esta parte da configuração oferece ao utilizador a capacidade de executar operações essenciais da loja, como a gestão geral da loja, por exemplo o ponto de venda (Point Of Sale - POS) que pode ser uma caixa registadora, e a Gestão de stock. Configurar, visualizar e manter produtos, preços e promoções ao nível da loja são importantes para muitos retalhistas. No retalho típico, é basicamente aqui que gerem as suas empresas diariamente. O POS é uma ferramenta essencial para a gestão da empresa em tudo o que envolva o atendimento ao cliente, com ele o funcionário pode pagar as faturas do cliente. Pode-se ver este tipo de ferramentas em cafés, restaurantes, pastelarias, entre outros. (Faria, 2021)

A funcionalidade do Backoffice da loja foi projetada para simplicidade e controlo de processo. Quando a equipa entra no sistema, eles só vêem produtos e documentos relevantes para sua função/loja, para garantir a execução correta e produtiva do processo. Os dispositivos móveis são suportados por todos os processos da loja para oferecer mobilidade, simplicidade e trabalho liderado por processos.

É no Frontoffice que se entra em contacto diretamente com o cliente, para que uma parte muito vital de uma operação de Retalho seja tratada e mantida por essa parte. O Omni

Channel, baseia-se no uso de diferentes canais estando eles interligados entre si para facilitar a comunicação (Fonseca, 2021), oferecendo diferentes pontos de compra e interação com o cliente. Hoje, uma loja virtual é apenas mais uma loja na sua configuração, um Mobile POS, como por exemplo as caixas registradoras do supermercado. A gestão de clientes é tratada com a máxima confiabilidade (LSRetail, 2020).

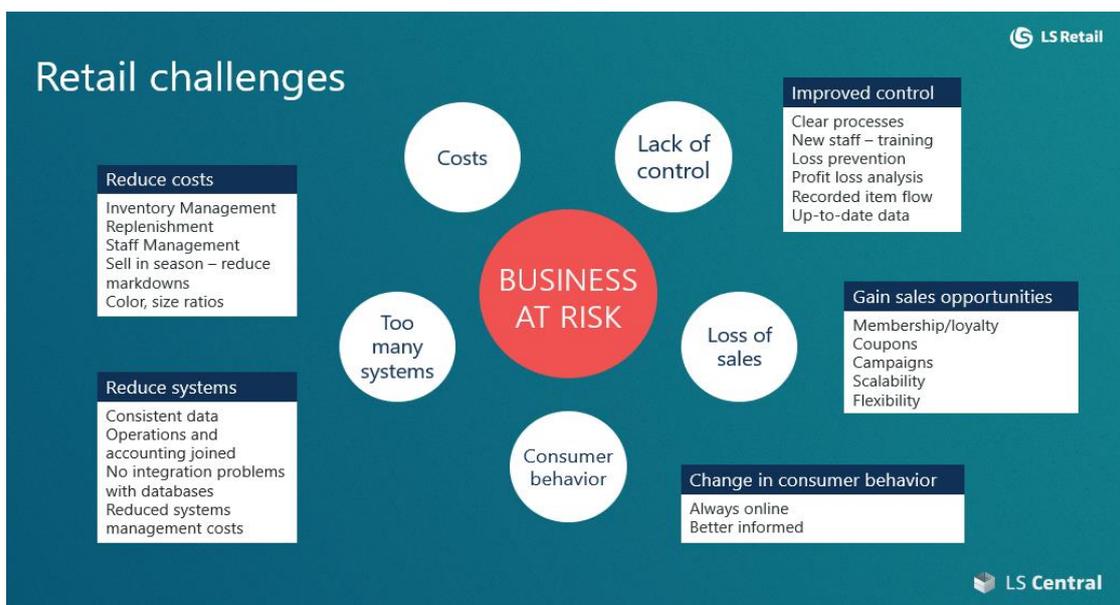


Figura 4: Desafios do LS Retail. adaptado de LS Retail, 2020.

1.3 - Motivação e Objetivos

A motivação deste trabalho reside em aprimorar o *software* atual que se encontra numa tecnologia antiga, onde a ideia passa pela alteração para uma nova tecnologia acrescentando novas funcionalidades para assim prestar um melhor serviço aos seus clientes, fazendo com que estes voltem à loja para compras futuras.

O objetivo global deste trabalho foca na mudança de uma solução já existente na versão Microsoft BC que se encontra na linguagem de C/AL (Microsoft, Microsoft, 2021) para a nova versão na linguagem de AL (Microsoft, Programming in AL, 2021). Juntamente com isto será também necessário fazer integração com o B2BOptic LensCatalog (SF6) e por último será necessário o desenvolvimento de uma aplicação para o cliente final com a particularidade de incidir no setor das óticas. Finalmente o *software* será integrado com empresas parceiras.

Tendo em vista o tema proposto para este trabalho, para o título de Mestre em Engenharia Informática é necessário ter em conta os atos de Engenharia relacionados com informática e sistemas de informação. Conhecimentos científicos e tecnológicos

indispensáveis para recorrer aos princípios de Engenharia no planeamento, projeto e implementação de sistemas, processos e serviços de informação e dos recursos informáticos a eles associados. Em concreto, as suas atividades abrangem: (1) a análise de negócio e engenharia de requisitos; (2) a conceção, construção, teste e validação de soluções informáticas; (3) o planeamento e gestão de infraestruturas de tecnologias de informação; (4) a gestão de projetos de sistemas de informação; e (5) o planeamento e auditoria de sistemas de informação.

Neste trabalho é fundamental ter em atenção os principais atos acima referidos, uma vez que são parte fulcral para o desenvolvimento do mesmo.

1.4 - Organização do Documento

No capítulo dois é apresentada informação sobre ERP'S, quais as vantagens e desvantagens do seu uso e alguns exemplos. É também apresentada alguma informação sobre o LS RETAIL ligado as óticas e efetuada uma descrição geral sobre o Business Central, mostrando diversos exemplos de objetos, é também feita uma breve apresentação sobre o formato SF6. No final do capítulo 2, é apresentado um estudo sobre os diferentes tipos de teste de *software*, de que forma é que estes devem ser realizados e os diferentes tipos que existem.

No capítulo 3 são apresentados os requisitos funcionais de cada uma das partes desenvolvidas, diagramas de casos de uso, quais tipos de *software* usados, como foi feito o desenvolvimento e, finalmente, são apresentados os testes que foram realizados ao longo do seu desenvolvimento.

No capítulo 4 são apresentadas as conclusões do trabalho realizado e algumas sugestões de trabalho futuro.

1.5 - O Estágio

O estágio ocorreu na Arquiconsultnearshore, nos seus escritórios em Vila Real. Iniciou com a integração numa equipa de trabalho, constituída por pessoas com grande espírito de entreajuda e com as quais havia reuniões semanais para fazer pontos de situação do trabalho desenvolvido até à data que envolvia a apresentação de vídeos e imagens que refletiam os desenvolvimentos realizados.

Para o desenvolvimento deste trabalho houve um período de formação de 4 meses, de aprendizagem das plataformas NAV e BC 365. Neste período foi iniciada a formação na linguagem AL, importante no processo da migração do *add-on*. A formação foi importante para a adquirir as bases essenciais para dar início ao desenvolvimento da tarefa, sendo que a maior aprendizagem foi obtida ao longo do desenvolvimento do trabalho.

Para a componente da aplicação móvel, não houve formação específica, tendo sido necessário um período de conhecimento e aprendizagem do funcionamento da mesma, para posteriormente poder passar ao processo de adição das funcionalidades de acordo com a construção inicial.

No início do estágio foram definidos dois desafios, que consistiram em uma migração para a linguagem AL de um *add-on* das óticas já existente, que se encontrava na linguagem de C/AL, e a adaptação de uma aplicação móvel do LS Retail, adicionando novas funcionalidades relacionadas com os serviços de vendas das óticas. No primeiro caso o pretendido foi criar as mesmas funcionalidades existentes na versão antiga, mas desta vez usando o cliente *browser*. Com isto, o funcionário da loja pode encomendar os óculos ou outro tipo de produtos relacionados, passar faturas, fazer pagamentos sem sair do *browser*. Na segunda componente do trabalho, com a alteração da aplicação do LS Retail o pretendido foi criar novas funcionalidade para permitir ao cliente comprar os seus óculos com a respetiva graduação e a escolha das hastes e do tipo de lentes, sem ter de sair de casa, usando a mesma aplicação que usa para comprar outros produtos, como por exemplo produtos alimentares ou até roupa.

2 - Revisão do Estado da Arte

O Planeamento dos recursos a Empresa, mais conhecido como ERP, contém diversos sistemas e aplicações que são usadas por empresas no seu dia-a-dia para fazer uma gestão integral do seu trabalho. O ERP, permite que a empresa controle desde atividades comerciais, administrativas, vendas, ou seja, oferece suporte para as atividades operacionais da empresa. A sua utilização potencia a comunicação entre os vários processos do negócio, resultando daí uma perspectiva melhoria dos resultados operacionais da empresa.

O ERP não se aplica só a grandes empresas. Este adapta-se a todas as empresas, independentemente do tamanho e do tipo. Como grande parte das empresas possuem áreas críticas que necessitam de uma maior atenção na sua gestão, o ERP consegue dar essa ajuda, porque possui soluções integradas para otimizar, automatizar e vincular os diversos departamentos, conseguindo assim uma sintonia entre todos eles.

Pode dizer-se que o ERP é um sistema que possibilita um fluxo de informação único, contínuo e consistente por toda a empresa, sob uma única base de dados, como podemos ver na figura 5 Sendo este um sistema que permite desenhar um amplo cenário para os negócios da empresa (Chopra, 2003).

Uma das vantagens destes sistemas vem da facilidade com que estes sistemas partilham dados entre si. O Microsoft Dynamics 365 Business Central – BC 365 apresenta uma configuração rápida e uma utilização simples. Para a implementação de funcionalidades específicas de setor comprovadas relevantes para as suas necessidades, este ERP preenche os requisitos tornando-se uma ferramenta que ajuda os seus utilizadores a serem mais eficazes.

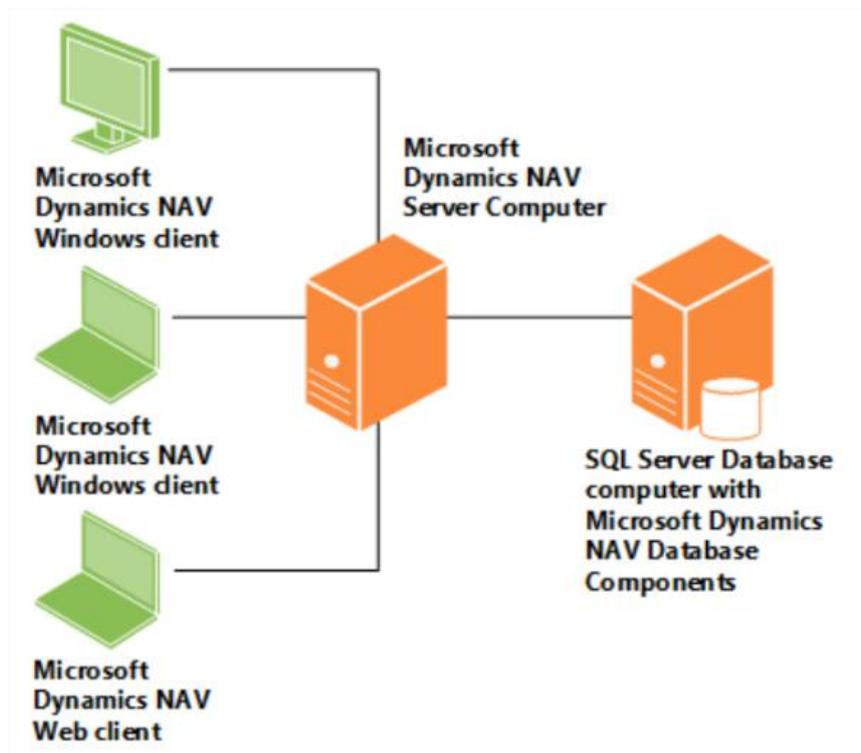


Figura 5: Estrutura do Microsoft Dynamics Nav, de (Brodie, 2017)

2.1 - Vantagens e Desvantagens de um Sistema ERP

Com o uso de *software* de ERP, existe uma infinidade de benefícios que podem ser obtidos através da sua implementação. A implementação de um ERP abrange todo o espectro de atividades e une todos os processos em toda a organização, figura 6, este vem conduzir à otimização de operações comerciais sendo capaz de melhorar a eficiência e impulsionar o crescimento do negócio. Contudo os sistemas ERP possuem vantagens e desvantagens na implementação dos diversos sistemas, variando estas consoante os diferentes autores.

(STAIR & REYNOLDS, 2006) acreditam que as vantagens principais da implementação do ERP são: a eliminação de sistemas ineficientes, a simplificação da adoção de processos de trabalho aperfeiçoados, a melhoria no acesso aos dados para a tomada de decisões operacionais e a padronização tecnológica. Como desvantagens, apontam: as despesas e tempo de implementação, mudanças na integração com outros sistemas e riscos ao adotar apenas um fornecedor.

Já os autores (Turban, Rainer, & Potter, 2005) mostram como benefícios da integração dos sistemas: redução de *stock*, redução de trabalhadores, aumento da produtividade, melhor gestão de pedidos, melhorias no ciclo financeiro, redução de custos de tecnologia de informação, redução de custos de aquisição, melhor gestão de caixa, aumento dos lucros, redução de custos de transporte e logística, redução da manutenção e cumprimento dos prazos de entrega; por outro lado, como desvantagens apresentam a visibilidade das informações, novos processos e mais complexos, melhor reação do consumidor, padronização, flexibilidade, globalização e desempenho do negócio.

Consoante (Simkin, 2014) afirma que as principais vantagens são: integração das diversas funções; eliminação de dados e sistemas redundantes; possibilidade de integração com fornecedores e clientes; permanentes melhorias do sistema ao longo da evolução tecnológica; acesso à informação em tempo real, melhorando o processo de tomada de decisão; aumento da produtividade e da eficiência de processos internos.

Ao nível das desvantagens do sistema ERP (Moura, 2006) relata que os sistemas ERP, na realidade, não são capazes de cobrir todas as funções de todas as empresas e, por essa razão, é sempre necessário integrar outros sistemas de informação com o ERP, diminuindo a capacidade de adaptação em ambientes de mudança contínua.

De uma forma geral os sistemas ERP apresentam diversas vantagens e têm muita popularidade junto das empresas, mas também possuem desvantagens no que diz respeito a sua adaptação nas diferentes organizações, com requisitos específicos.

Por último, a adoção deste tipo de tecnologia para uma empresa pode trazer ganhos significativos, mas caso a escolha da tecnologia não seja a correta, ou seja, não vá de encontro aos seus requisitos, pode levar a uma ineficácia de processos.



Figura 6: Exemplos de ERP, adaptado de (PRO, 2020)

2.2 - Tendências de ERP

O ERP teve de se adaptar rapidamente à evolução rápida do mercado, pois existem cada vez mais novas tecnologias no mercado e outras são retiradas porque passam a ser obsoletas. Para que um ERP não se transforme numa tecnologia obsoleta, tem de acompanhar a evolução tecnológica. Para isso integrou-se com o armazenamento em nuvem que ajudou a complementar falhas existentes em relação ao armazenamento de dados e o seu acesso. O “cloud computing” introduziu soluções mais baratas, mas rápidas e poderosas para o ERP (Santos, 26).

Com a mobilidade, o ERP conseguiu que o setor das vendas tivesse acesso aos dados em qualquer lugar e em qualquer altura, não sendo necessário a existência de um escritório, isto veio de encontro com a nova realidade dos clientes que cada vez mais fazem compras com os dispositivos móveis. (Santos, 26) A Internet das Coisas, ou IoT (Internet of Things), diz respeito a dispositivos inteligentes e ligados entre si, capazes de monitorizar e gerir a produtividade de uma forma autónoma e automatizada. A IoT é capaz de fornecer objetos que conseguem distribuir dados em uma rede sem a necessidade de interação humana. Este tipo de tecnologia tem vindo a ganhar espaço no mercado da indústria e deve ser expandido para todo o mercado levando a uma integração desses equipamentos em um ERP mais abrangente. A IoT é um salto de

produtividade, e o ERP com essa capacidade é um avanço de gestão. É uma tecnologia capaz de se tornar uma solução estratégica completa (Barrett, 2018).

Outra vantagem do ERP ir para a nuvem é que a sua implementação se torna mais simples assim como o retorno de dados é muito mais rápido. Essa tendência ficou conhecida como ERP modular. Em vez de adquirir uma solução num pacote fechado, a IT contrata a tecnologia como um serviço, podendo adicionar ou retirar funções e ferramentas de acordo com as suas necessidades (Alecrim, 2020).

2.3 - Descrição Geral do Produto – Microsoft Dynamics 365 Business Central

O Microsoft Dynamics 365 Business Central é uma solução de negócio multilinguagem e multi-moeda utilizada para a gestão dos seus recursos por diversas empresas em todo mundo

O Microsoft Dynamics 365 partilha a mesma base de código que o NAV onde possui uma interface com o utilizador, figura 7, que é fácil de utilizar, de aprendizagem rápida e focado no aumento de produtividade, permitindo assim que os seus colaboradores tenham tempo para realizar outras tarefas, ajudando no aumento da produtividade da empresa.

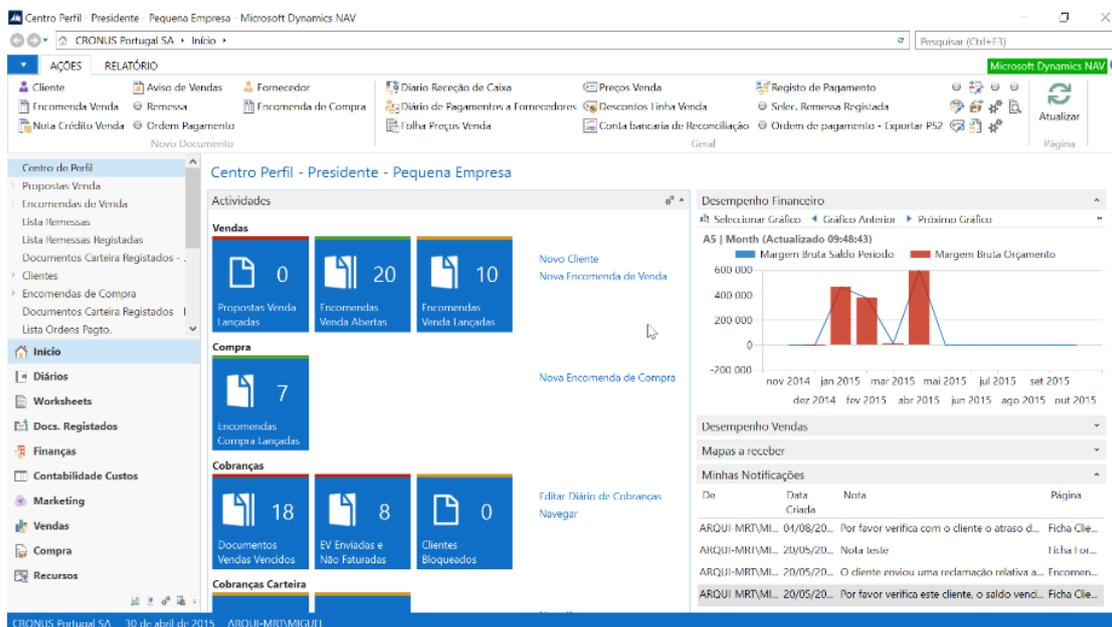


Figura 7: Centro perfil, Microsoft Dynamics Nav.

Trata-se de uma solução escalável que cresce e evolui com as necessidades da empresa, acessível nas mais variadas plataformas, desde *desktop*, *browser*, *sharepoint* e dispositivos móveis, disponibilizando desta forma soluções que se adaptam a todos os perfis de utilização (ver figura 8 e figura 9).

Cliente Web:

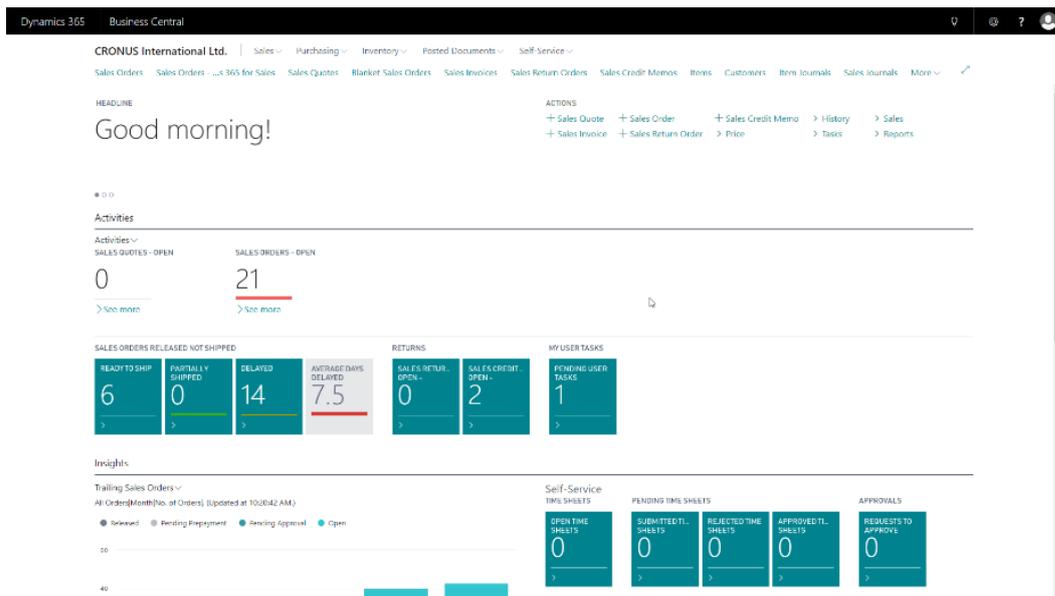


Figura 8: Cliente web

Cliente Tablet:

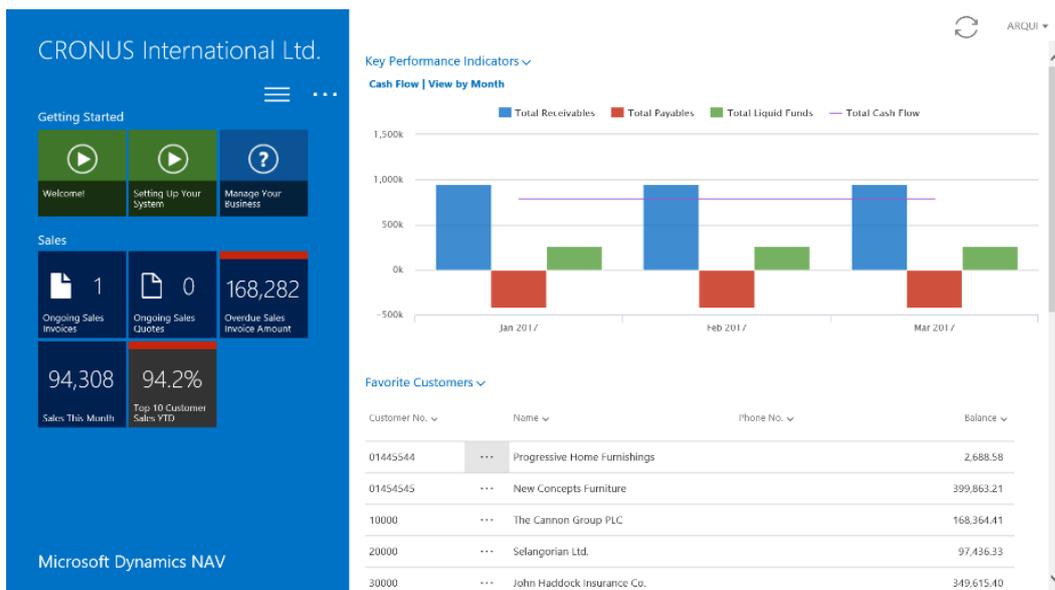


Figura 9: Cliente Tablet.

A integração nativa com SQL Reporting Services, Excel e Power BI confere uma capacidade analítica e de reporting, permite a criação de faturas de acordo com os dados do sistema, bem como outros documentos.

O Microsoft Dynamics 365 Business Central tem como principal foco na interação com o utilizador o aumento da produtividade, servindo-se para isso dos conceitos já aplicados nas outras soluções da Microsoft, como o Office, facilitando desta forma a aprendizagem e utilização das diferentes funcionalidades. Este benefício é visível no conceito de Role Center que disponibiliza a cada utilizador o conjunto de funcionalidades e ferramentas que são mais úteis para o desempenho da sua atividade e responsabilidades específicas dentro da organização (Business Central, 2020).

2.3.1 - Visão Tecnológica

O Microsoft Dynamics 365 Business Central compreende uma arquitetura de três camadas (3 tier) com alta escalabilidade e robustez, separando a base de dados, a lógica de negócio e a interface com o utilizador em níveis independentes. A camada de base de dados é suportada em Microsoft SQL Server, como podemos ver na figura 9 (Brodie, 2017).

A lógica de negócio está isolada na segunda camada, *service tier*, disponibilizando um ambiente de desenvolvimento, configuração e integração com outras aplicações via *web services* que permite responder a um elevado leque de requisitos de negócio e técnicos.

Finalmente a interação com o utilizador é feita através de uma solução de *rendering* que permite a disponibilização da solução em qualquer plataforma, seja *desktop*, *web*, *sharepoint* ou através de aplicações para dispositivos móveis (Business Central, 2020).

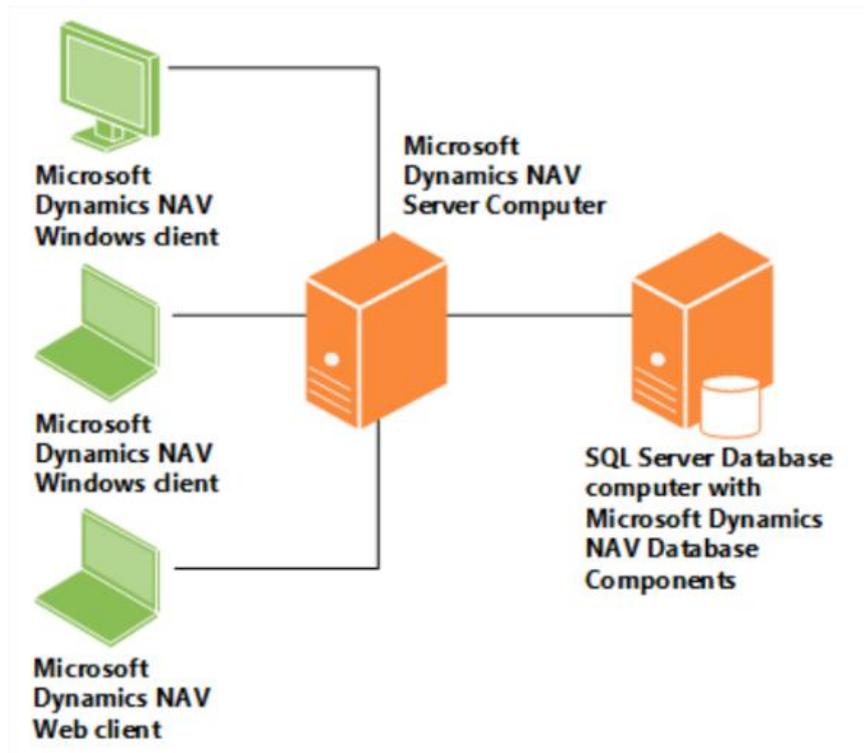


Figura 10: Estrutura do Microsoft Dynamics NAV, de (Brodie, 2017)

O Microsoft Dynamics 365 Business Central está preparado nativamente para ser instalado em Microsoft Azure, aumentando desta forma o leque de opções em termos de ambiente, na medida em que permite ao cliente escolher entre *cloud*, com a garantia que a aplicação está preparada para esta vertente, ou *On Premises* (What is On Premises?, 2021) figura 10 (Business Central, 2020).

Microsoft cloud for business



Figura 11: Cloud para Microsoft.

O ambiente de desenvolvimento, exemplificado na figura 11, totalmente integrado e que utiliza linguagem e conceitos proprietários, permite adaptar e/ou expandir a lógica de negócio a requisitos específicos da empresa através de ferramentas de alto nível, como por exemplo o Microsoft Azure e o Office 365, que aceleram e facilitam as customizações, trata-se duma ferramenta que atua na camada de lógica de negócio e, por isso, é totalmente independente da interface com o utilizador, permitindo desta forma disponibilizar de forma imediata as alterações em todas as plataformas de acesso ao Microsoft Dynamics 365 Business Central.

```
Pharmacy_NO] - Microsoft Dynamics NAV Development Environment - [Codeunit 10000705 LS Retail helper - C/AL Editor]
File Edit View Tools Window Help
142 END;
143
144 InvoiceRound(CurrencyCode : Code[10];Amount : Decimal) : Decimal
145 SalesSetup.GET;
146
147 IF SalesSetup."Invoice Rounding" THEN
148 BEGIN
149     IF CurrencyCode = '' THEN
150         BEGIN;
151             GeneralLedgerSetup.GET();
152             GeneralLedgerSetup.TESTFIELD("Inv. Rounding Precision (LCY)");
153             InvoiceRoundingPrecision := GeneralLedgerSetup."Inv. Rounding Precision (LCY)";
154             InvoiceRoundingType := GeneralLedgerSetup."Inv. Rounding Type (LCY)";
155         END
156     ELSE
157         BEGIN;
158             Currency.GET(CurrencyCode);
159             Currency.TESTFIELD("Invoice Rounding Precision");
160             InvoiceRoundingPrecision := Currency."Invoice Rounding Precision";
161             InvoiceRoundingType := Currency."Invoice Rounding Type";
162         END;
163
164     CASE InvoiceRoundingType OF
165         InvoiceRoundingType::Nearest:
166             Amount := ROUND(Amount,InvoiceRoundingPrecision,'=');
167             InvoiceRoundingType:=Up;
168             Amount := ROUND(Amount,InvoiceRoundingPrecision,'>');
169             InvoiceRoundingType:=Down;
170             Amount := ROUND(Amount,InvoiceRoundingPrecision,'<');
171         END;
172     END
173 ELSE
174     Amount := ROUND(Amount);
175
176 EXIT(Amount);
177
178 AmountRound(CurrencyCode : Code[10];Amount : Decimal) : Decimal
179 IF CurrencyCode = '' THEN
180 BEGIN;
181     GeneralLedgerSetup.GET();
182     InvoiceRoundingPrecision := GeneralLedgerSetup."Amount Rounding Precision";
183     IF InvoiceRoundingPrecision = 0 THEN
184         InvoiceRoundingPrecision := 0.01;
185     END
186 ELSE
187     BEGIN;
```

Figura 12: Ambiente de desenvolvimento NAV.

A integração com aplicações externas é facilitada através da disponibilização de *webservices*, figura 12, que permitem expor a quase totalidade da lógica de negócio, sempre sujeita a níveis de segurança e permissões, de forma a interagir e a construir fluxos de troca de dados com aplicações externas. O motor de disponibilização dos *webservices* é, em grande parte, configurável o que diminui a dependência tecnológica. Adicionalmente são disponibilizados um conjunto de *webservices out-of-the-box* como por exemplo a integração com serviços de taxas de câmbio, serviços de OCR, *feeds* de ODATA, (ODATA, 2021) ou serviços de reconhecimento por OCR.

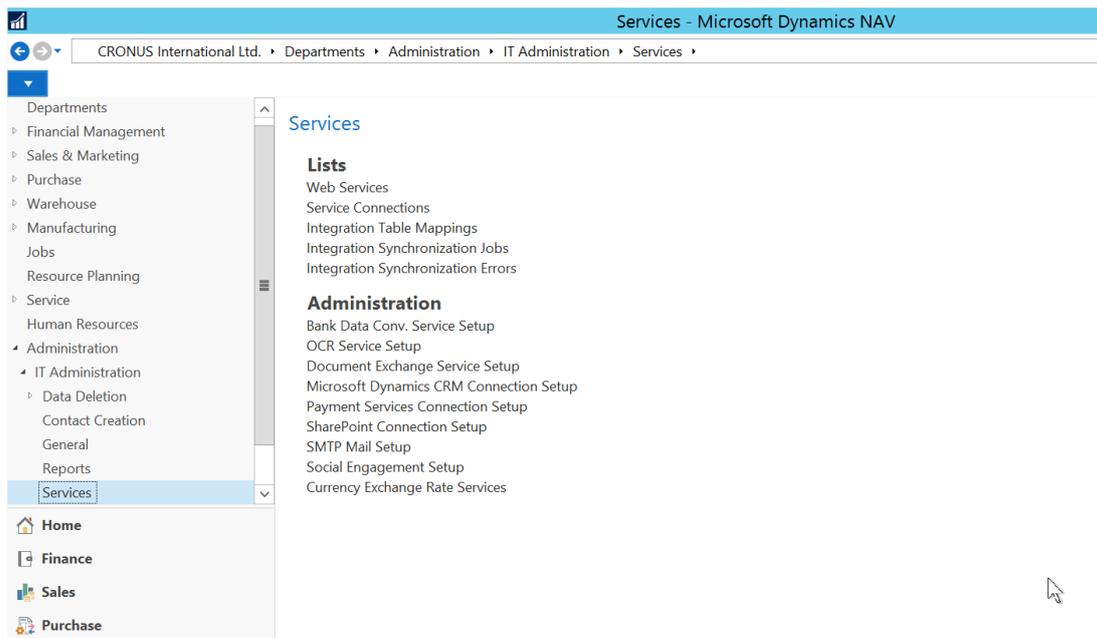


Figura 13: Ambiente de cliente NAV.

A definição de utilizadores pode ser efetuada de várias formas aproveitando as definições existentes no *Active Directory*. Os utilizadores são agrupados em perfis (ver figura 13), de forma a serem disponibilizadas as funcionalidades relevantes para a sua atividade. São definidas as permissões através da associação a grupos de permissões, como por exemplo a permissão de “super user” que, por sua vez, indicam as ações que cada utilizador pode efetuar.

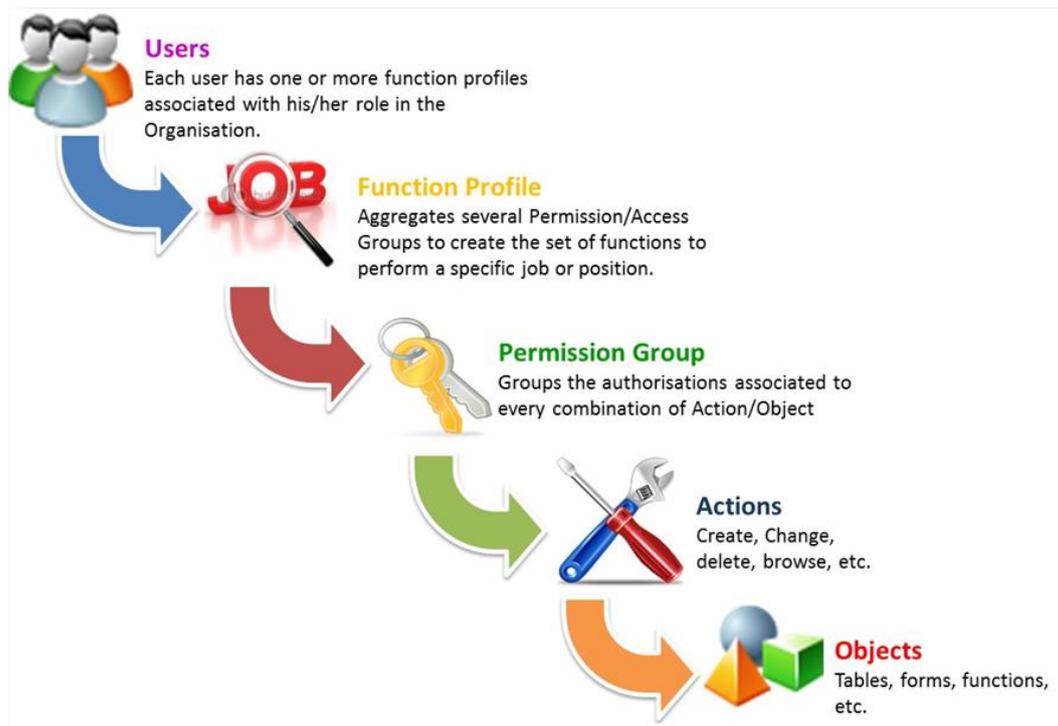


Figura 14 :Imagem ilustrativa de ferramentas disponibilizadas pelo NAV.

O Microsoft Dynamics 365 Business Central disponibiliza um conjunto de ferramentas que permitem auditar e controlar os registos e iterações com a aplicação, a configuração pode ir ao nível do campo definindo para quais é importante registar a inserção, modificação ou eliminação. Sempre que a configuração do registo de alterações está ativa o Microsoft Dynamics 365 Business Central guarda quem, quando e o que foi modificado de forma a permitir e facilitar as atividades de *audit trail*.

2.4 - LS RETAIL – Ótica

LS Retail fornece soluções simples para a gestão empresarial. As soluções são altamente configuráveis e ajudam retalhistas e hoteleiros, a otimizar práticas de negócio, aumentando as receitas e a satisfazer novos e antigos clientes, com isto é possível adaptar o *software* às necessidades de cada empresa (Arquiconsult, 2020).

As empresas têm como principal objetivo a ambição de crescimento, encarando novas maneiras de negócio de uma forma flexível e cooperativa. É neste ponto que se enquadra as tecnologias Microsoft Dynamics. A Arquiconsult faculta soluções baseadas na capacidade de Backoffice do Microsoft Dynamics, fazendo em parceria com os seus

clientes e assumindo, em muitos casos, alguns custos de desenvolvimento quando daí resultem produtos comercializáveis, permitindo assim uma redução de investimento do cliente e contribuindo assim para uma comunicação mais eficaz entre as pessoas, tornando a informação disponível em qualquer lugar, em qualquer momento (LSRetail, 2020).

2.4.1 - Visão Geral das Óticas

O Dynamics Óticas é um produto complementar ao Microsoft Dynamics 365 Business Central e ao LS Retail que começou a ser desenvolvida em 2005. O Dynamics Óticas é um *add-On* ao BC365. Com a junção das funcionalidades das óticas e as funcionalidades gerais do BC365, as empresas conseguem fazer uma gestão total das suas funcionalidades, desde as mais simples às mais complexas.

Este *software* possui 3 grandes funcionalidades gerais, são elas Headoffice, Backoffice e POS. Este também controla o fluxo do trabalho com o CRM, no setor de compras e otimização das regras de negócio.

2.5 - Desafios na Área do Retalho

A área do retalho apresenta vários desafios e para que ela se torne sem falhas e seja cada vez mais uma área de confiança e transparente, é necessário que esta tente corresponder às necessidades dos clientes. Com o decorrer dos anos, a demanda de produtos pelos clientes foi aumentando chegando a um patamar nunca tão alto como agora. Para tal, a área do retalho evoluiu e com o avanço da indústria e da tecnologia proporcionou aos clientes uma vasta gama de opções deixando estes de ter apenas as tradicionais escolhas. O cliente é peça integrante para a área do retalho, seja este, uma única pessoa ou uma grande empresa, o principal objetivo é o foco das suas necessidades. Estas necessidades estão em constante mudança, o que torna um desafio acompanhar estas constantes mudanças e proporcionar pequenas inovações no produto para que este continue a cativar o cliente (LS Retail, 2020).

O conhecimento do público-alvo permite a criação de uma dependência com os clientes. Para que esta dependência permaneça ao longo do tempo é necessário que o cliente não perca o interesse e acabe por terminar o negócio, o que dificulta a crescimento na área do retalho.

A complexidade da gestão dentro da área do retalho vai depender sempre do tamanho da empresa. No entanto, existem outros três desafios menos óbvios que sempre devem ser considerados, mas às vezes são esquecidos:

- Escalabilidade - Aumentar a escala de algo equivale a equipas e fluxos de comunicação mais complexos. É necessário otimizar a interação entre funcionários e gerentes;
- Automação - Sinais automatizados são importantes para garantir que os membros da equipa sejam responsabilizados por suas ações, para ajudá-los a organizar suas tarefas e para resolver problemas, em vez de apenas criar interrupções.
- Relatórios - É crucial medir os resultados e verificar se as equipas estão atingindo os seus objetivos. Essa é uma tarefa especialmente importante para os gerentes de operações, pois permite que eles tomem decisões mais bem informadas, com base em uma revisão das comunicações atuais e futuras.

Tudo isso pode garantir que a execução e a conformidade estejam sempre em níveis ideais.

Na área do retalho a procura de melhores soluções está iminente e, para tal, os retalhistas procuram soluções automatizadas para simplificar o seu processo de negócios. Muitas empresas estão a consciencializar-se nas questões ambientais, sociais e éticas, o que se reflete nas decisões na opção de escolha pelos consumidores. Sendo que as pessoas estão cada vez mais a comprar, não só com base na qualidade do produto, mas também nas práticas comerciais que os produtores empregam.

Uma maneira fácil e impactante de se tornar mais sustentável é reduzir a utilização de papel. Por exemplo, a substituição do papel por uma solução de *software* para planear suas operações ou pode o envio de faturas e recibos por correio eletrónico. Mudar para opções de energia limpa, a fim de reduzir o desperdício e a reciclagem, também são grandes passos para se tornar sustentável (LS Retail, 2020).

2.6 - SF6

O SF6 é um formato de documentos que tem como objetivo oferecer uma melhor representação possível das gamas de produtos disponíveis nos fabricantes de lentes. (Specification Lens Product Catalogue. (s.d.))

O formato SF6 foi verificado quanto à sua funcionalidade nos principais mercados europeus e foi alinhado com os padrões DIN e ISSO, sendo que possui diferentes ficheiros que possuem dados relativos a cada empresa.

Todos os códigos básicos de lente que ocorrem nos arquivos LensPrice.Dat, Combination.Dat, LensRange.Dat, LensGeo.Dat, Information.Dat e OeCodes.Dat, devem ser definidos em LensType.Dat. Todos os códigos de opção que ocorrem nos arquivos OptionsColor.Dat, OptionsPrice.Dat, Combination.Dat, LensRange.Dat, Information.Dat e OeCodes.Dat devem ser definidos em Options.Dat. Todos os índices de opção usados no arquivo OptionsPrice.Dat devem ser definidos em LensType.Dat. Todos os índices de intervalo usados no arquivo LensPrice.Dat devem ser definidos em LensRange.Dat. Os nomes dos produtos fornecidos nos arquivos LensType.Dat, Options.Dat e OptionsColor.Dat devem ser exclusivos em cada arquivo, apesar de não fazer parte da chave primária.

Para evitar qualquer confusão entre a gama de produtos do SF6, as abreviaturas de cada fabricante são registadas no documento Head.Dat. Estas devem ser exclusivas para cada fabricante e utilizador. A tabela 1 mostra as abreviações já usadas para os vários fabricantes de lentes.

Tabela 1 - Abreviações usadas pelos fabricantes de lentes

Abbreviation	Manufacturer
AO	American Optical
EMO	Emmerich Optik
ESS	Essilor
HLD	Hoya Lens
IN	Indo
KNE	Knecht
MMS	Metzler Mailshop
NI	Nikon
NIK	Nika
NOH	Nordhorn Optik
NOR	Norville
NOV	Novacel
OGE	Optik Ges.m.b.H
OPH	Ophthalmica
OPT	Optovision
OSW	Optiswiss (CH)
PEN	Pentax
REI	Reize Optik (CH)
R+H	Rupp & Hubrach
ROD	Rodenstock
SCE	Schulz
SCH	Schneider Optische Werke
SEI	Seiko
SAO	Signet Armorlite
SO	Sola
SV	Starvision
STR	Stratemeier
TOE	TOE
WET	Wetzlich
CZ	Zeiss (D, Lux)

Tendo em conta a gama de ficheiros do formato SF6 e as especificações, estes são detalhados nos seguintes ficheiros.

Head.Dat - Informações gerais sobre o fabricante e a lista de preços, bem como a definição do grupo de cilindros e prismas

LensType.Dat - Especificações de lentes básicas. Os produtos são identificados por um único código do fabricante.

LensRange.Dat - Faixa de produção e disponibilidade de produtos e combinações de produtos.

LensGeo.Dat - Descreve a geometria de cada lente, incluindo o ponto de referência e a localização da linha.

LensPrice.Dat - Preços básicos das lentes para faixas e opções disponíveis.

Options.Dat - Especifica extras opcionais nos grupos de produção e cor. Os produtos são identificados com um código de fabricante exclusivo.

OptionsColor.Dat - Especificações de cores individuais para cada produto colorido.

OptionsPrice.Dat- Preços para extras em Options.dat, bem como para faixas mais altas e prismáticas.

Combination.Dat - Informações sobre restrições relativas à combinação de extras opcionais.

Information.Dat - Informações adicionais sobre o produto.

OeCodes.Dat - Códigos Erfa e OPC para lentes e revestimentos.

O código do país é determinado de acordo com a ISO 3166-1 ALPHA-2.

O SF6 permite também ter dados específicos do fabricante.

É possível que em certos casos as especificações acima descritas do cliente não permitam descrever toda a gama de produtos de um fabricante. Nesses casos, é possível integrar um formato de documento específico do fabricante (File Specification Lens Product Catalogue).

O SF6 tem desvantagens tais como:

- Ambíguo (em gerenciamento e avaliação de dados)
- Interpretação complexa
- Redundante (mesma informação em arquivos diferentes)
- Combinações complexas não são possíveis
- Lacunas na documentação -> levam a diferentes interpretações

2.7 -Testes de Software

Neste capítulo é abordado o tema dos testes de *software* que são aplicados para testar a qualidade e problemas que possam ocorrer, ou seja, verificar se o *software* cumpre todas as funcionalidades pedidas pelo cliente. Mas antes de percebermos o que são os testes, necessitamos de compreender o que é a qualidade de *software*. Dependendo do tipo de autores pode ter diversos significados.

Segundo o artigo de Kitchenham e Pfleeger, sobre a qualidade do *software*, ele mostra de forma explícita em que consiste o conceito, de cinco pontos de vista diferentes: o ponto de vista transcendental, o ponto de vista do utilizador, o ponto de vista da produção, o ponto de vista do produto e, por fim, o ponto de vista baseado em valores (B. Kitchenham and S. L. Pfleeger, Jan. 1996).

O ponto de vista transcendental aponta a qualidade como algo que pode ser reconhecido, mas difícil de definir. É um ponto de vista que não diz respeito apenas aos testes de *software*, mas também a outras áreas complexas do dia-a-dia.

O segundo ponto de vista propõe que existe realmente qualidade se a função do produto final satisfizer as necessidades do utilizador.

No ponto de vista da produção, a qualidade é medida em conformidade com as especificações iniciais.

No ponto de vista do produto, a qualidade é inerente às características do produto, ou seja, as qualidades internas de um produto determinam as qualidades externas.

O ponto de vista baseado em valores a qualidade de um *software* está relacionado com o preço que alguém está disposto a pagar por ela.

As características mais gerais, como reutilização de código, eficiência, entre outras podem ser subdivididas várias vezes até termos uma característica do *software*. Um exemplo disso é o desempenho de um *software* é uma característica muito genérica, que é subdividida no modelo ISO 25000 em Comportamento de Tempo, Utilização de Recursos e Capacidade. (ISO, 2019) Mesmo essas características precisam ser subdivididas em diferentes critérios, sendo um exemplo de um critério o tempo de execução de uma função (Técnicas).

No conceito de qualidade de *software* os mais influentes são ISO 9126 (o International Organization for Standardization) (Stefani & Xenos, 2008) (Paulk, 2002). Um grupo de especialistas propôs sob a tutela da ISO, diferentes modelos, onde primeiro modelo propõe seis categorias independentes relativas à qualidade: funcionalidade, fiabilidade, usabilidade, eficiência, sustentabilidade e portabilidade.

Outro modelo o CMM, mostra que um processo de desenvolvimento é avaliado numa escala de zero a cinco, onde o nível um corresponde ao mais baixo e o nível cinco corresponde ao mais alto de maturidade, isto é, o nível otimizado.

Para os testes de *software*, existem dois modelos bastante conhecidos, o TPI (Test Process Improvement) (Andersin, 2004) e o TMM (Testing Maturity Model) (Burnstein, 1999). Os dois modelos permitem a qualquer empresa verificar o estado atual dos testes de *software*.

Os testes de *software* são feitos para encontrar defeitos, erros na aplicação. Estes podem ser feitos ao longo do seu desenvolvimento, para detetar pequenos erros

para que mais tarde não se venham a tornar grandes problemas. Com estes testes é possível também verificar se a aplicação compre os requisitos iniciais e se vai de encontro ao pedido do cliente, ou seja, se faz aquilo que é pretendido.

2.7.1 - Como Realizar um Teste de Software

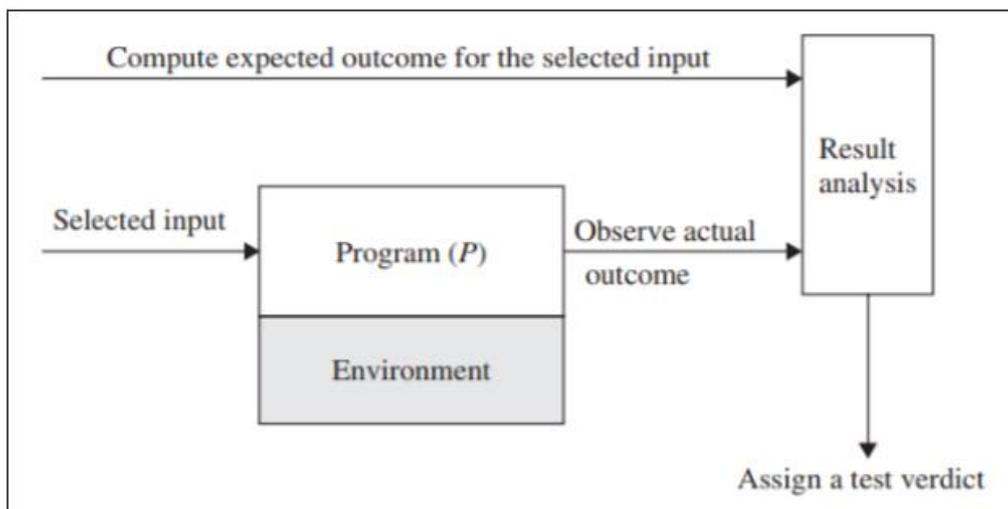


Figura 15: Exemplo da realização de testes, adaptado de (Naik, 2011).

Para a realização de testes no *software* inicialmente é necessário definir qual é o resultado final. Deve-se criar diferentes tipos de situações para garantir que o programa vai funcionar. Devemos escolher as entradas para o teste (inputs) que levem à obtenção do nosso objetivo final (figura 14). De seguida, deve-se ter o programa para realizar a execução. Devemos ativar tudo o que seja necessário para que o teste possa ocorrer, como por exemplo conexão à base de dados. Quando estiver tudo pronto, o código é executado e deve-se esperar pelos resultados obtidos. Depois de termos os resultados, estes devem ser analisados para garantir que o resultado é o esperado e, caso não seja, deve ser feita uma análise e verificar o que falhou. Por exemplo, se tivermos uma calculadora e somarmos dois números inteiros, o resultado esperado é um valor inteiro e caso este seja um decimal, podemos concluir que algo está errado e deve ser resolvido, mas caso seja um inteiro, então temos um resultado coerente com o esperado, como podemos ver pela análise da figura 14, onde é selecionado um valor de entrada e depois de ser aplicado o programa é feita a análise do valor de saída e é analisado o resultado, para garantir que está de acordo com o esperado.

2.7.2-Tipos de Testes de Software

Os testes de *software* são executados em diferentes níveis, segundo (Naik, 2011), para garantir que todo o sistema é testado durante o seu ciclo de vida desde o seu planeamento até deixar de ser usado. O *software* por norma, passa por 4 fases de testes sendo estes testes unitários, testes de integração, e testes de aceitação e interface.

Por norma os testes unitários, integrados e de interface são executados por diferentes pessoas dentro da empresa, enquanto que os de aceitação são executados pelo cliente.

A figura 15 é representa o diagrama dos diferentes testes.

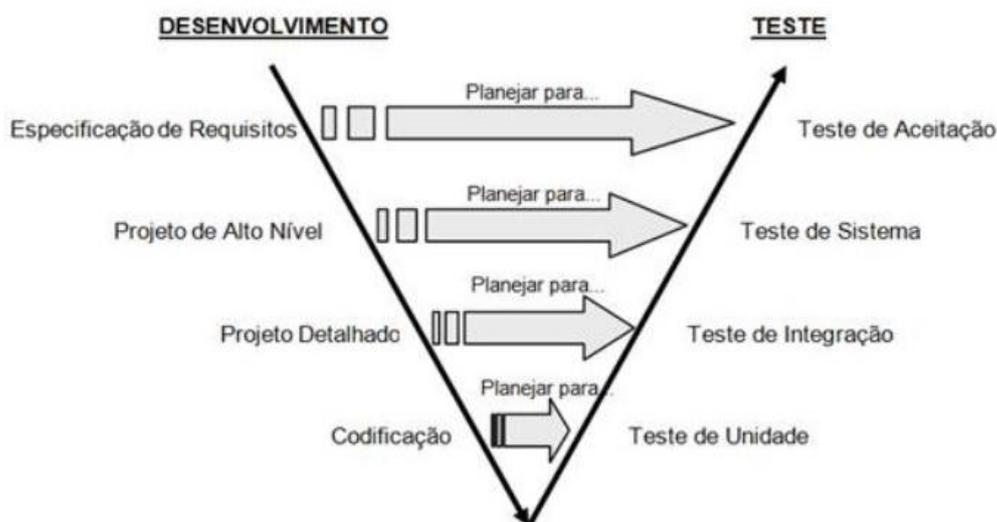


Figura 16:Exemplo de testes de software, adaptado de (Caetano).

Com a análise da figura 14, podemos ver que os testes unitários, são executados por norma pelos programadores, em que estes executam parte do código, por exemplo funções, para perceber se o resultado final é o esperado. Quando o resultado é positivo então estas pequenas partes de código testadas podem se agrupar a um conjunto que dará o resultado final do sistema. Ao contrário dos testes unitários os testes de integração por norma são executados por um conjunto de engenheiros de *software* e, estes devem ser sempre que possível, engenheiros de integração de sistemas, para garantir que o sistema é um sistema estável, capaz de suportar o rigor dos testes de sistema. Estes incluem uma vasta gama de testes, como a segurança, a robustez,

funcionalidade, carga, estabilidade, stresse, performance e confiabilidade. É por isso que os testes de integração são tão importantes no desenvolvimento de *software*. Os testes de sistema incluem testes de usabilidade, testes de carga, testes de regressão, testes de recuperação, testes de migração, testes funcionais e testes ao *hardware* e *software*.

Em conjunto, estes 3 tipos de testes perfazem os testes de regressão, isto é, um subnível de testes que são executados sempre que um componente importante do sistema sofre uma alteração, com o objetivo de garantir que nenhuma parte do sistema foi afetada por uma alteração.

Por último depois do *software* ter passado pelos diferentes testes e o resultado ser positivo, o produto final é entregue ao cliente.

3 - Protótipo Implementado

Neste capítulo é feita uma apresentação sobre as funcionalidades do sistema e a análise de requisitos, onde é apresentado aquilo que o cliente pretende que o seu sistema faça, para ajudar em loja. Com esta análise o programador consegue perceber o que futuramente o sistema vai fazer, o que irá ajudar nos testes finais, que iremos falar mais a frente no subcapítulo 3.8.

3.1.1 - Arquitetura do Sistema



Figura 17: Arquitetura do sistema. (Retail, 2021)

O utilizador em loja inicialmente recebe os dados do cliente e faz uma pesquisa por ele ou então caso seja um novo cliente este deve ser adicionado a base de dados, de seguida é criada a *glass sales order*, para fazer a encomenda, depois de esta estar concluída é feita a compra dos óculos, quando tiverem todo o material é feita a montagem e por fim é feita a entrega ao cliente, como podemos ver na figura 16.

3.1.2 - Análise de Requisitos da Migração

O sistema deverá permitir:

- 1 - pesquisar por cliente;
- 2 - criar uma ordem de venda de óculos;
- 3 - escolher a data de levantamento da encomenda.
- 4 - escolher o tipo de lentes;
- 5 - escolher o vendedor;
- 6 - escolher o tipo de material das lentes;
- 7 - escolher o tipo de encomenda;
- 8 - escolher as hastes;
- 9 - escolher a lente direita;

- 10 - escolher a lente esquerda;
- 11 - escolher a configuração das lentes;
- 12 - escolher a cor;
- 13 - terminar encomenda;
- 14 - efetuar pagamento;
- 15 - passar fatura ao cliente;
- 16 - fazer um pré-pagamento;
- 17 - importar dados através do SF6;

3.1.3 – Diagrama de Casos de Uso da Migração

No diagrama de Casos-de-Uso da extensão é feita uma apresentação dos casos de uso possíveis desde a criação da fatura até ao seu pagamento. O funcionário da loja pode criar a ordem de venda de óculos com as diferentes características dependendo das necessidades do cliente, este pode fazer a pré-configuração das lentes, escolher lentes e as hastes e pode fazer total configuração dos Óculos, o funcionário pode ainda criar a venda, cria a respetiva encomenda, pesquisar por clientes, se o cliente pretender pode fazer um pré-pagamento através de vales, no final permite fazer o respetivo pagamento total e passar a fatura ao cliente.

Para que o funcionário possa ver todas as encomendas feitas até ao momento é possível ver o histórico de todas as suas vendas em loja.

O sistema permite também importar dados através das normas SF6 (wiki.b2boptic, 2021), para cada loja.

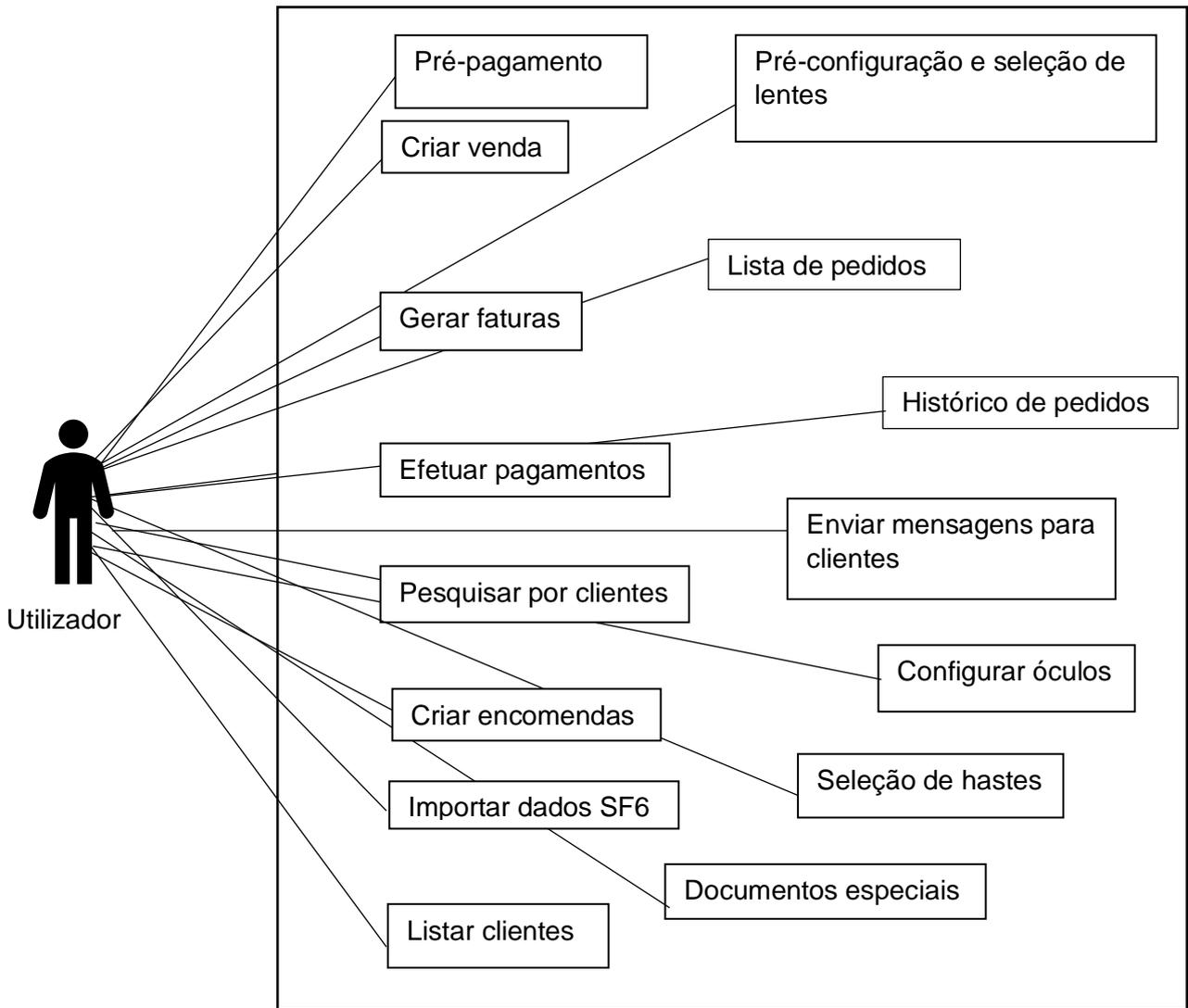


Diagrama de casos de uso da Migração

3.2 - Implementação da Extensão

Nesta parte do meu trabalho mostro o meu trabalho prático que é iniciado pela migração de C/AL para AL. De seguida foi feita o SF6, que permite importação de dados para a base de dados de diferentes lojas. Por último foi criado uma aplicação móvel para que o cliente pode-se fazer a compra dos seus óculos sem ter de sair de casa.

3.2.1. Migração de BC para AL

Na fase inicial do meu trabalho comecei por fazer uma conversão de uma extensão para uma nova versão do ambiente de desenvolvimento do Business Central 365, para a linguagem em AL. A versão antiga encontrava-se na linguagem de C/AL. Ou seja, na versão antiga o código que pertencia apenas às óticas encontrava-se nos objetos *standard*, o que causa problemas sempre que seja necessário fazer alterações para novas versões e implica que estejamos sempre a alterar os objetos *standard*. Com a migração isso deixa de acontecer todo o código que diz respeito apenas a extensão das óticas passa para a versão AL, onde são criados objetos que são extensões de objetos da versão *standard*, assim o código *standard* mantém-se inalterado. Para isso é necessário criar extensões de tabelas e de páginas sempre que seja necessário adicionar campos ou funções novas a estas. No caso de *codeunits*, como estas não possuem extensão apenas se pode usar a subscrição de eventos, o mesmo acontece quando é necessário acrescentar código numa função que já exista nas tabelas e nas páginas. Quando não é possível é necessário criar objetos novos, em parte cópia dos antigos, mas com o código novo.

3.3 – Software usado na Migração

Nesta parte do trabalho foi utilizado o SQL Server para restaurar a base de dados e para a passagem de dados, o Visual Studio Code para o desenvolvimento de AL e o C/AL, para ver os objetos que tinham de ser migrados.

3.3.1-Structured Query Language

Structured Query Language - SQL, é uma linguagem de programação desenvolvida para lidar com base de dados relacionais sendo que, foi inspirada na álgebra relacional. Tendo como primeiro desenvolvimento no início dos anos 70 nos laboratórios da IBM dentro do projeto System R cujo objetivo era demonstrar a viabilidade da implementação do modelo relacional proposto por E.F. Cood.

Tendo esta linguagem sido desenvolvida ao longo do tempo, torna-se numa ferramenta viável e de simples utilização para o utilizador. Esta linguagem diferencia-se de outras linguagens de consulta de base de dados, sendo declarativa e no sentido em que uma consulta SQL especifica a forma de resultado e não o caminho para chegar a ele. Esta foi desenvolvida para que os utilizadores pudessem aceder e modificar os dados de uma empresa em simultâneo, de maneira descomplicada e unificada (Microsoft SQL Server, 2020).

A programação em SQL pode ser usada para a criação de base de dados, análises ou criação de tabelas, seguindo principalmente os comandos apresentados na tabela 2.

Tabela 2 - Comandos SQL

Comando	Descrição
CREATE	Criação da base de dados ou tabelas
DROP	Apagar tabelas da base de dados
ALTER TABLE	Alterar tabelas na base de dados
INSERT	Inserir registos na base de dados
DELETE	Apagar dados na base de dados
UPDATE	Atualizar registos na base de dados
SELECT	Consulta de dados da base de dados

Principais comandos para a criação/utilização de uma base de dados:

1- O comando para a criação da base de dados e/ou tabelas é o CREATE e apresenta a seguinte sintaxe.

CREATE DATABASE Nome da base de dados: Este comando é usado sempre que é necessário criar uma base de dados nova.

CREATE TABLE Nome da tabela: Este comando é usado sempre que é necessário criar uma nova tabela em uma base de dados.

2- O comando para apagar tabelas da base de dados é o **DROP**.

DROP TABLE Nome da tabela: O comando **ALTER TABLE** é utilizado para modificar uma tabela já criada.

ALTER TABLE Nome da tabela **ADD** dados Tipo de dados

ADD é o comando para adicionar um campo, pode também ser substituído **DELETE**, pelo para eliminar e o **MODIFY** para modificar;

Dados são os valores que serão inseridos na tabela. Estes campos devem estar por ordem e separados por vírgulas.

3- O comando para inserção na base de dados é o **INSERT**.

INSERT INTO Nome da tabela **VALUES** (lista de campos)

VALUES (lista de dados)

Os dados pertencentes á lista de dados devem estar pela mesma ordem que é descrita na lista de campos, todos separados por vírgula. Se for utilizado o comando **SELECT** o mesmo deve retornar a mesma quantidade de colunas com os mesmos tipos de dados que estavam na lista de campos.

4- O comando para apagar dados é o **DELETE**:

DELETE FROM Nome da tabela

WHERE condição

WHERE: impõe uma condição sobre a execução do comando, por exemplo apenas apagar um campo com um determinado identificador (ID).

5- O comando para atualizar registos é o UPDATE, que tem a seguinte sintaxe:

UPDATE Nome da tabela

SET CAMPO = “valor a atualizar”

WHERE condição

O comando para efetuar subconsultas é o SELECT acoplada com uma instrução SELECT, INSERT, DELETE ou UPDATE.

SELECT * FROM Nome da tabela **WHERE** Campo = valor

O SQL sendo a ferramenta utilizada pela empresa para criação/manutenção de base de dados foi escolhida para o desenvolvimento deste projeto. A sua utilização passou por efetuar o restauro da base de dados e a inserção de dados na mesma.

3.3.2 - POWERSHELL

Esta ferramenta foi criada pela Microsoft, sendo que é uma Shell que potencia o acesso a vários serviços de um sistema operativo. Pode ter como base a linha de comandos ou incluir uma interface gráfica do utilizador.

O PowerShell do Windows é a *shell* desenvolvida pela Microsoft, baseada no CLR (.NET framework common Language RunTime) e o .Net Framework que inclui uma linha de comandos e uma linguagem de script. Esta *shell* de linha de comandos do Windows foi especialmente projetada para administradores de sistema, incluído um *prompt* interativo e um ambiente de script que pode ser usado independentemente ou em combinação (What is PowerShell?, 2020).

Baseando-se no conceito de “command-let”, este permite ao powershell executar tarefas complexas sendo que os comandos mais usuais se encontram listados abaixo, na tabela 3:

Tabela 3 - Alguns comandos PowerShell

Cmdlet	Function
Get-Location	obter o diretório atual
Set-Location	alterar o diretório atual
Copy-Item	copiar arquivos
Remove-Item	remover um arquivo ou diretório
Move-Item	mover um arquivo
Rename-Item	renomear um arquivo
New-Item	criar um novo arquivo vazio ou diretório

Neste projeto o powershell teve uma utilização fundamental visto que permitiu a exportação de objetos nos quais foi efetuado um *split* e posterior conversão de CAL para AL.

3.3.3 - Visual Studio Code

O Visual Studio Code - VSC é um editor de código-fonte desenvolvido pela Microsoft para os vários sistemas operativos Windows, macOS e Linux. Sendo um programa de *open source* permite e facilita a criação de extensões e novas funcionalidades.

É um editor extremamente leve e multiplataforma, bastante fácil de usar pelos seus utilizadores, este pode ser usado como ferramenta de programação ou como um simples editor de texto.

Sendo que, neste projeto a utilização do VSC permitiu efetuar a migração de NAV Dynamics Business Central – BC para linguagem AI. A sua utilização foi essencial para a publicação dos packages e para fazer os devidos testes necessários neste projeto.

Esta linguagem de programação usada para a manipulação de dados tal como recuperar, inserir e modificar registos de uma base de dados do Dynamics 365 Business Central, permitindo o controlo da execução de vários objetos como páginas, *reports* e *codeunits* (Visual Studio Code, 2020).

Grande parte dos objetos no Dynamics 365 Business Central contêm eventos onde o programador pode adicionar seu código AL. Existem eventos para os seguintes objetos:

1. Tabelas e extensões de tabelas
2. Campos da tabela
3. Páginas e extensões de páginas
4. Reports
5. Data items
6. XMLports
7. Queries

As tabelas servem para criar campos e estas serão criadas na base de dados para guardar os respetivos dados. Nas páginas são criados os *layouts* para serem apresentados ao cliente. Nos *reports* são criados os *layouts* para imprimir as respetivas faturas. Os XMLPorts são usados para importar e exportar dados.

3.3.3.1 - Diretrizes para colocar o código AL

Como qualquer projeto que seja bem-sucedido é necessário seguir um conjunto de padrões e regras, como tal na criação de código em AL deve seguir algumas regras recomendadas.

Para que o projeto tenha maior rentabilidade e permita um melhor uso de código e aconselhável no futuro escrever-se código em pequenas partes, facilitando um *design* limpo e possibilitando a reutilização de código. Por exemplo, caso seja necessário colocar código num objeto em vez de este ser colocado numa *codeunit*, deve ser colocado em eventos relativos a funções, ou a campos da tabela.

3.3.3.2- Reutilização de código e regras da criação de objetos

A reutilização de código torna o desenvolvimento de aplicativos mais rápido e fácil. Se a organização do código AL estiver dentro dos padrões sugeridos, as suas aplicações estarão menos propensas a erros. Ao centralizar o código, o risco de criar inconsistências involuntariamente executando, por exemplo, em vários *triggers* que possuem o mesmo campo de tabela que sua expressão de origem. Numa alteração de

código, poderá acontecer, por lapso, o esquecimento de alguns desses *triggers* ou mesmo cometer um erro na sua modificação.

Neste projeto a utilização do AL serviu para criação de novos objetos, tanto tabelas, como codeunits, reports, xmlPorts e pages.

Inicialmente para criar o projeto no VSC foi necessário criar o package, para assim se ter acesso aos objetos standard do BC, para isso precisamos de ligar o projeto ao serviço de BC 365 e à base de dados. Posto isto, e já com o package criado é possível a criação de tabelas, seguindo um conjunto de regras.

1. Primeiramente é necessário definir os campos, como por exemplo figura 17.

```
field(1; "Entry No."; Integer)
{
    Caption = 'Entry No.';
}
```

Figura 18: Criação de fields.

2. Definir quais os campos chaves, como por exemplo figura 18.

```
keys
{
    - reference
    key(Key1; "Entry No.")
    {
    }
}
```

Figura 19: Definição de chaves primárias.

3. Caso seja pertinente a criação de *triggers* para fazer alterações na tabela e *procedures*, como por exemplo figura 19 e 20.

```
trigger OnDelete()
var
    ProcessPermCommentLine: Record 70089;
begin
    ProcessPermCommentLine.RESET;
    ProcessPermCommentLine.SETRANGE("Entry No.", "Entry No.");
    ProcessPermCommentLine.DELETEALL;
end;
```

Figura 20: Exemplo de trigger.

```
procedure SetSubstantiationText(var ProcessPermissionEntry: Record 70087; Value: Text)
var
    DataOutputStream: OutStream;
    SubstantiationText: BigText;
begin
    CLEAR(ProcessPermissionEntry.Substantiation);
    SubstantiationText.ADDTEXT(Value);
    ProcessPermissionEntry.Substantiation.CREATEOUTSTREAM(DataOutputStream);
    SubstantiationText.WRITE(DataOutputStream);
end;
```

Figura 21: Exemplo de procedure.

Neste projeto, foram encontradas dificuldades a nível de tabelas, sendo que, a mais trabalhosa e que requereu especial atenção foi a tabela 7500, cujo nome é “b2bOptic Export Setup”, onde o seu código após a conversão revelou ser mais complexo e com mais correções a efetuar.

A conversão automática de objetos de BC para AL, revelou não efetuar uma conversão perfeita o que implica que haja alguns erros que devem ser corrigidos no AL. Os principais erros desta conversão passaram por simples detalhes como aspas (”) nos ID dos objetos, sendo que quando uma variável num objeto é definida todas estas aspas tem de ser retiradas, pois com elas, os objetos não são reconhecidos, como por exemplo figura 21, 22 e 23.

```
var
    ExcelBuf: Record "370";
    CityandStreet: Record 70078;
```

Figura 22: Definição de variáveis globais.

Também é necessário colocar “”, em nomes com mais de uma palavra, para que seja reconhecido como um só nome e não vários.

```
field(2; Post Code; Code[20])
{
```

Figura 23: Exemplo de erro em nomes.

```
12 references
field(2; "Post Code"; Code[20])
{
```

Figura 24: Exemplo de correção de erro.

Para criar tabelas *extention*, isto é, para acrescentar alterações nas tabelas *standard* é necessário criar extensões de tabelas para se poder acrescentar campos nas tabelas e, criar novos *procedures*. Para poder acrescentar código em *triggers* de campos, ou em *procedures* é necessário usar os eventos existentes para poder acrescentar as alterações. Em caso em que não haja eventos, os casos tornam-se mais complexos e devem ser analisados caso a caso.

A tabela “Sales Line” é exemplo de uma tabela *extention*, onde existiam novos campos para acrescentar e tinha um *procedure* que era necessário acrescentar código, visto que não era necessário a criação de novos eventos, procedeu-se a elaboração de um novo *procedure* com o código, onde este foi chamado em prol do outro.

Para a criação de novas *pages*, é necessário definir a *sourcetable*, ou seja, a tabela que tem como base aquelas páginas. Para a criação do *designer* tem de se definir o *layout* e os grupos, onde, dentro do *layout* se coloca os campos que vão aparecer na página, como por exemplo figura 24 e 25.

```
Caption = 'Discount Action';  
PageType = Card;  
SourceTable = "Discount Action";
```

Figura 25: Exemplo de criação de pages.

```
group(Allgemein)  
{  
  0 references  
  field(Code; Code)  
  {  
  }  
  0 references  
  field(Description; Description)  
  {  
  }  
  0 references  
  field("Customer Price Group"; "Customer Price Group")  
  {  
  }  
  0 references  
  field("Customer Disc. Group"; "Customer Disc. Group")  
  {  
  }  
}
```

Figura 26: Exemplo de inserção de campos em pages.

Na página deve-se também escolher o tipo de página, isto é, se é *card* ou *list*.

Pode-se também acrescentar partes às páginas tendo por base uma nova tabela, como por exemplo figura 26.

```
part(JJ; "Discount Action Valid")
{
    SubPageLink = "Discount Action Code" = FIELD(Code);
}
```

Figura 27: Definição de part pages.

Nas páginas pode-se também criar *actions*, para despoletar determinadas ações, como por exemplo figura 27.

```
actions
{
    0 references
    area(embedding)
    {
        0 references
        action(Kunden)
        {
            Caption = 'Customers';
            Image = Customer;
            RunObject = Page "Store Cust. List";
        }
    }
}
```

Figura 28: Exemplo de criação de actions.

Também podemos colocar código nos *trigger* da página e *procedures*.

Algumas páginas de ótica que foram criadas continham variáveis .net, este tipo de variáveis deixou de ser suportado pela linguagem AL, sendo então necessário criar um ficheiro com algumas destas variáveis para ser possível o seu uso, é também necessário fazer a importação da respetiva DLL, para que esta possa ser reconhecida.

As *page extention* servem para adicionar novas funcionalidades às páginas *standard*, onde é possível adicionar campos, *actions* e novos *procedures*.

Para inserir código nos métodos existentes deve ser usado os eventos, como por exemplo o *OnOpenPageEvent*, que serve para adicionar código ao *trigger* *OnOpenPage*.

De seguida convertem-se as *codeunits* tanto da *standard* como as novas de ótica, onde nas *codeunits* encontram-se *procedures* que efetuam alteração de dados, entre outras coisas.

Depois de ser efetuada esta conversão houve necessidade de resolver os problemas em todos os objetos visto que ocorria o erro das aspas.

Algumas *codeunits* possuem variáveis .net, que não deram para resolver, sendo comentado esse código para que assim fosse possível publicar o package, tendo como exemplo a *codeunit* 71001, como por exemplo figura 28.

```
var
    64 references
    XMLDom: DotNet XmlDocument;
    33 references
    nodes: DotNet XmlNodeList;
    157 references
    node: DotNet XmlNode;
    0 references
    PHInitialized: Boolean;
    0 references
    ACInitialized: Boolean;
    /* QPH: DotNet DcModule;
    QAC: DotNet AcModule;*/
```

Figura 29: Exemplo de variáveis dotnet

Para as *codeunits standard* de BC e de LS em todas as que era necessário fazer alterações, onde foi criada uma nova *codeunit* para chamar os eventos de cada *procedure* e no caso de não haver eventos, foi feito uma análise de cada situação para resolver o problema.

Nos XmlPort, apenas foi necessário converter os objetos novos, estes são usados para o SF6, eles são usados para importar dados de ficheiros para a BD.

É neles que se define a forma como os dados são importados tendo de corresponder a forma como estes se encontram nos ficheiros, como por exemplo figura 29 e 30.

Nos objetos de XmlPort deve ser definido o formato do ficheiro, e todos os elementos que vão ser importados.

```
Caption = 'Import Lens Data Header';
Format = FixedText;
TextEncoding = WINDOWS;
```

Figura 30: Exemplo de Formato do import

```
schema
{
  0 references
  textelement(Root)
  {
    0 references
    tableelement("<lt>"; "Lens Type")
    {
      AutoReplace = false;
      AutoSave = false;
      AutoUpdate = false;
      XmlName = 'LT';
      1 reference
      textattribute(HstCBG)
      {
        Width = 6;
      }
    }
  }
}
```

Figura 31: Definição de campos no xmlport

Nos *Reports*, todos eles foram corrigidos e houve a necessidade de corrigir erros de conversão de nomes atribuídos a alguns campos. Os *reports* não possuem extensão, caso seja necessário acrescentar código num *standard*, isso não é possível, é necessário criar um novo e chamar o *report* novo, e vez do atual (Visual Studio Code, 2020).

3.3.4 – Microsoft Dynamics 365 Business Central

O Microsoft Dynamics 365 Business Central é um sistema ERP da Microsoft. Este *software* faz parte da gama de produtos Microsoft Dynamics veio substituir Microsoft Dynamics NAV. O BC365 tem por base o código do NAV e a linguagem C/AL do NAV. O C/AL tem vindo a ser ultrapassado pela linguagem AL.

Para a migração foi usado um projeto que se encontrava em NAV com um *add-on* de Dynamics Optial (DO) e LSRetail, que estava na linguagem C/AL.

Teve de ser realizada a migração do *add-on* das Óticas para a extensão na linguagem AL.

Ao longo desta parte do projeto, houve algumas dificuldades tais como a criação do package pois a base de dados que foi disponibilizada, não estava funcional, houve a necessidade de recuperar uma base de dados de *backup* e importar todos os objetos de Dynamics Optical nesta. Outra grande dificuldade sentida foi na alteração de funções já existentes, em situações em que estas não possuíam eventos.

Outra situação que provocou dificuldade ao longo do desenvolvimento é o facto de a linguagem AL não suportar variáveis .net. (Business Central, 2020).

3.4-Implementação do SF6

Para a importação de todos os dados relativos aos diferentes ficheiros, foi necessário criar 22 xmlPort's diferentes. Isto foi necessário porque o formato a importar é diferente entre clientes, sendo que alguns ficheiros possuem conteúdo diferente.

Em cada um deles, é necessário definir quais são os campos a importar, para que assim os dados sejam importados para a base de dados corretamente e sem falhas, pois se houver campos a mais ou a menos tanto no ficheiro .txt, como no xmlPort, dá erro, os campos podem vir a vazio, ou seja, sem qualquer valor, mas tem de estar referenciados no ficheiro.

Na figura 31, podemos verificar que é definido o formato do documento a importar, também são definidos dentro do table elements os atributos, que são os campos que se encontram no documento .txt. Nos atributos é definido o tamanho dos campos, para que assim ele saiba ir buscar o campo correto, pela ordem que esta representada no ficheiro.

```
xmlport 70001 "Import Lens Type SF6.10.1"
{
  Caption = 'Import Lens Type';
  Format = FixedText;
  TextEncoding = WINDOWS;
  UseRequestPage = false;

  schema
  {
    0 references
    textelement(Root)
    {
      0 references
      tableelement("<lt>"; "Lens Type")
      {
        AutoReplace = false;
        AutoSave = false;
        AutoUpdate = false;
        XmlName = 'LT';
        0 references
        textattribute(HstCBG)
        {
          width = 6;
        }
      }
    }
  }
}
```

Figura 32: Exemplo de um documento xmlPort em AL.

Na figura 32 temos um exemplo de um documento onde podemos ver os campos que se encontram a vazio estão igualmente referenciados, mas a vazio. Na figura 33, temos um exemplo da importação dos dados do documento .txt para o programa.

A+2A AVON +Freeform 1.58 A TopCoat	31100100121600013041000000000001000010000000000010197	01000
A+2B AVON +Freeform 1.58 B TopCoat	31100100121600013041000000000001000010000000000010198	01000
A+2C AVON +Freeform 1.58 C TopCoat	31100100121600013041000000000001000010000000000010199	01000
A001L VPRO Freeform 1.50 L HMC	31100100121500013258000000000001000010000000000010326VPRO150	01000
A001M VPRO Freeform 1.50 M HMC	31100100121500013258000000000001000010000000000010325VPRO150	01000
A001XSVPRO Freeform 1.50 XS HMC	31100100121500013258000000000001000010000000000010324VPRO150	01000
A002L VPRO Freeform 1.60 L HMC	31100100121600013041000000000001000010000000000010329VPRO160	01000
A002M VPRO Freeform 1.60 M HMC	31100100121600013041000000000001000010000000000010328VPRO160	01000
A002XSVPRO Freeform 1.60 XS HMC	31100100121600013041000000000001000010000000000010327VPRO160	01000
A003L VPRO Freeform 1.67 L HMC	31100100121670013732000000000001000010000000000010332VPRO167	01000
A003M VPRO Freeform 1.67 M HMC	31100100121670013732000000000001000010000000000010331VPRO167	01000
A003XSVPRO Freeform 1.67 XS HMC	31100100121670013732000000000001000010000000000010330VPRO167	01000
A004 AVON Office Freeform 1.74 TopCoat	4110010012174001473300000000000100002000000000010533Office1.74	02000
A004L VPRO Freeform 1.74 L HMC	31100100121740014733000000000001000010000000000010335VPRO174	01000
A004M VPRO Freeform 1.74 M HMC	31100100121740014733000000000001000010000000000010334VPRO174	01000
A004XSVPRO Freeform 1.74 XS HMC	31100100121740014733000000000001000010000000000010333VPRO174	01000
A005L VPRO Freeform Trivex 1.53 L HMC	31100100141530011144000000000001000010000000000010338VPRO1.53	01000
A005M VPRO Freeform Trivex 1.53 M HMC	31100100141530011144000000000001000010000000000010337VPRO1.53	01000
A005XSVPRO Freeform Trivex 1.53 XS HMC	31100100141530011144000000000001000010000000000010336VPRO1.53	01000
A007L VPRO Freeform Pmax 1.60 L HMC	31100100141600013048000000000001000010000000000010341VPRO160 Pmax	01000
A007M VPRO Freeform Pmax 1.60 M HMC	31100100141600013048000000000001000010000000000010340VPRO160 Pmax	01000
A007XSVPRO Freeform Pmax 1.60 XS HMC	31100100141600013048000000000001000010000000000010339VPRO160 Pmax	01000
A101L VPRO Freeform SunTouch Braun 1.56 L HMC	31100100121560000000000000010000100001000000000010344VPRO156STB5	01000
A101M VPRO Freeform SunTouch Braun 1.56 M HMC	31100100121560000000000000010000100001000000000010343VPRO156STB5	01000
A101XSVPRO Freeform SunTouch Braun 1.56 XS HMC	31100100121560000000000000010000100001000000000010342VPRO156STB5	01000
A102L VPRO Freeform SunTouch Grau 1.56 L HMC	31100100121560000000000000010000100001000000000010347VPRO156STG5	01000

Figura 33: Exemplo de um documento no formato SF6.

The screenshot shows the SF6 software interface for document import. At the top, it displays 'LENS DATA | WORK DATE: 5/21/2019' and 'AMA · 2015-05-12'. Below this, there are navigation icons for 'Import Data', 'Set Status', 'Export', 'Show Attached', and 'More options'. The main section is divided into two tabs: 'Allgemein' and 'Head.Dat'. Under 'Allgemein', the 'Manufacturer Code' is set to 'AMA', 'Product Catalog' is 'SF6', 'Import Version' is '2015-05-12', and 'Status' is 'Open'. Under 'Head.Dat', the 'Version' is '6.10.2', 'Software ID' is 'ccgpl6edit-6.10.2.617', 'UID Manufacturer' is 'AMA--DE-DE-20150401-2.zip', 'Valid-from Date' is '4/1/2015', and 'Country Code ISO' is 'DE'. On the right side, there are sections for 'CYLINDER' and 'PRISM', each with 'Price Definition' set to '+' and 'Group' values of 4, 6, and 8.

Figura 34: Exemplo da importação do formato SF6, por loja

3.5 - Criação da Aplicação Móvel

Nesta parte do trabalho foram adicionadas novas funcionalidades a uma aplicação já existente do LS Retail Loyalty. Esta aplicação já permitia fazer compras de diferentes tipos de produtos, mas não possuía nada relacionado com as óticas.

A aplicação existente possui diferentes lojas das quais as pessoas podem escolher quando necessitam de levantar o produto em loja, Click&Collect, como por exemplo figura 34.

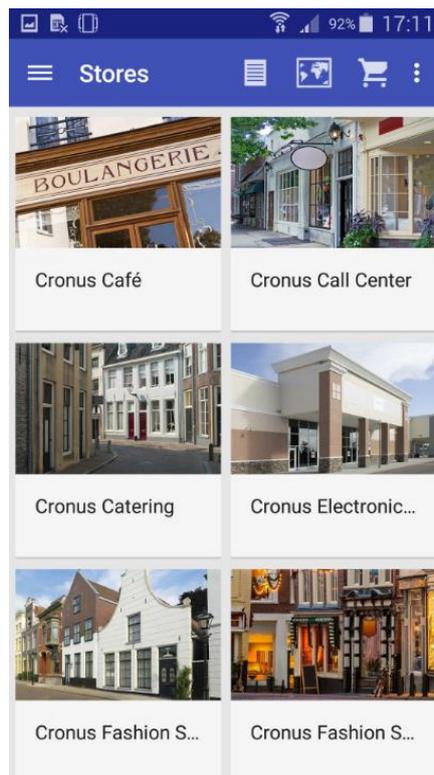


Figura 35: Exemplo de lojas existentes

Possui diversas categorias de produtos e, por consequência, diferentes grupos por categoria. O utilizador pode adicionar os produtos que desejar ao carrinho de compras, como por exemplo figura 35 e 36, e terminar a encomenda. Também é possível, verificar o histórico de encomendas criadas e guardar produtos nos favoritos.



Figura 36: Exemplo de algumas categorias

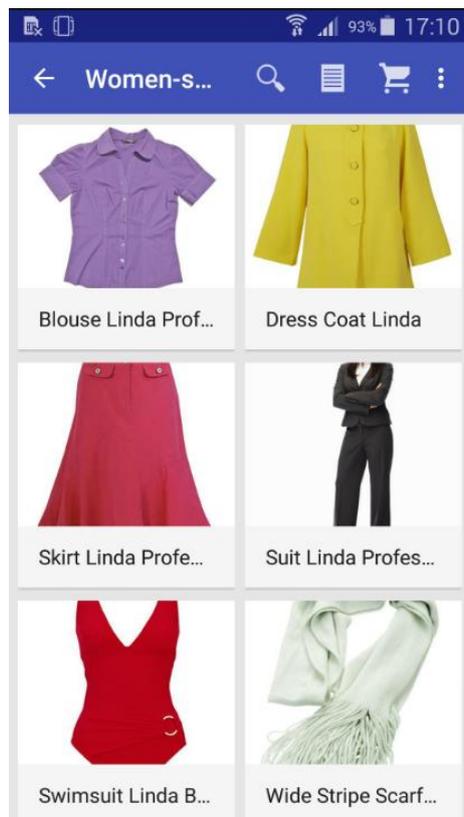


Figura 37: Exemplo de um ecrã da aplicação

O desafio foi criar do lado da APP o mesmo tipo de encomenda de óculos que existia do lado do Business Central, ou seja, o cliente estando em casa pode fazer a sua encomenda de óculos sem ter de se dirigir à loja.

3.5.1 - Análise de Requisitos da Aplicação LS Retail Loyalty

Nesta secção é feita uma apresentação sobre as funcionalidades da aplicação móvel e a análise de requisitos, onde é apresentado aquilo que o cliente final pode fazer com a aplicação sem ter de se deslocar a uma loja. Com esta análise o programador consegue perceber o que futuramente o sistema vai fazer, o que irá ajudar nos testes finais, que iremos falar mais a frente no capítulo 4

Na Lista abaixo é feita uma descrição de todas as possibilidades que o sistema deverá permitir realizar.

- 1- criar uma ordem de venda de óculos;
- 2- escolher o tipo de lentes;
- 3- escolher o vendedor;
- 4- escolher o tipo de material das lentes;
- 5- escolher o tipo de encomenda;
- 6- escolher as hastes;
- 7- escolher a lente direita;
- 8- escolher a lente esquerda;
- 9- adicionar os produtos ao carrinho;
- 10- terminar encomenda;
- 11- escolher o tipo pagamento;
- 12- escolher o tipo de levantamento da encomenda

3.5.2 - Diagrama de Casos de Uso

No diagrama de casos de uso da aplicação é feita uma apresentação dos casos de uso possíveis para a encomenda dos óculos na parte da APP. O cliente pode criar a ordem de venda de óculos com as diferentes características dependendo das suas necessidades. Este pode fazer a pré-configuração das lentes (material, tipo e graduação, pode também escolher o tipo de lente esquerda e direita) e as hastes e no final de tudo escolhido pode adicionar os produtos ao carrinho. Ao finalizar a encomenda este pode escolher a forma de pagamento, e o tipo de levantamento da encomenda.

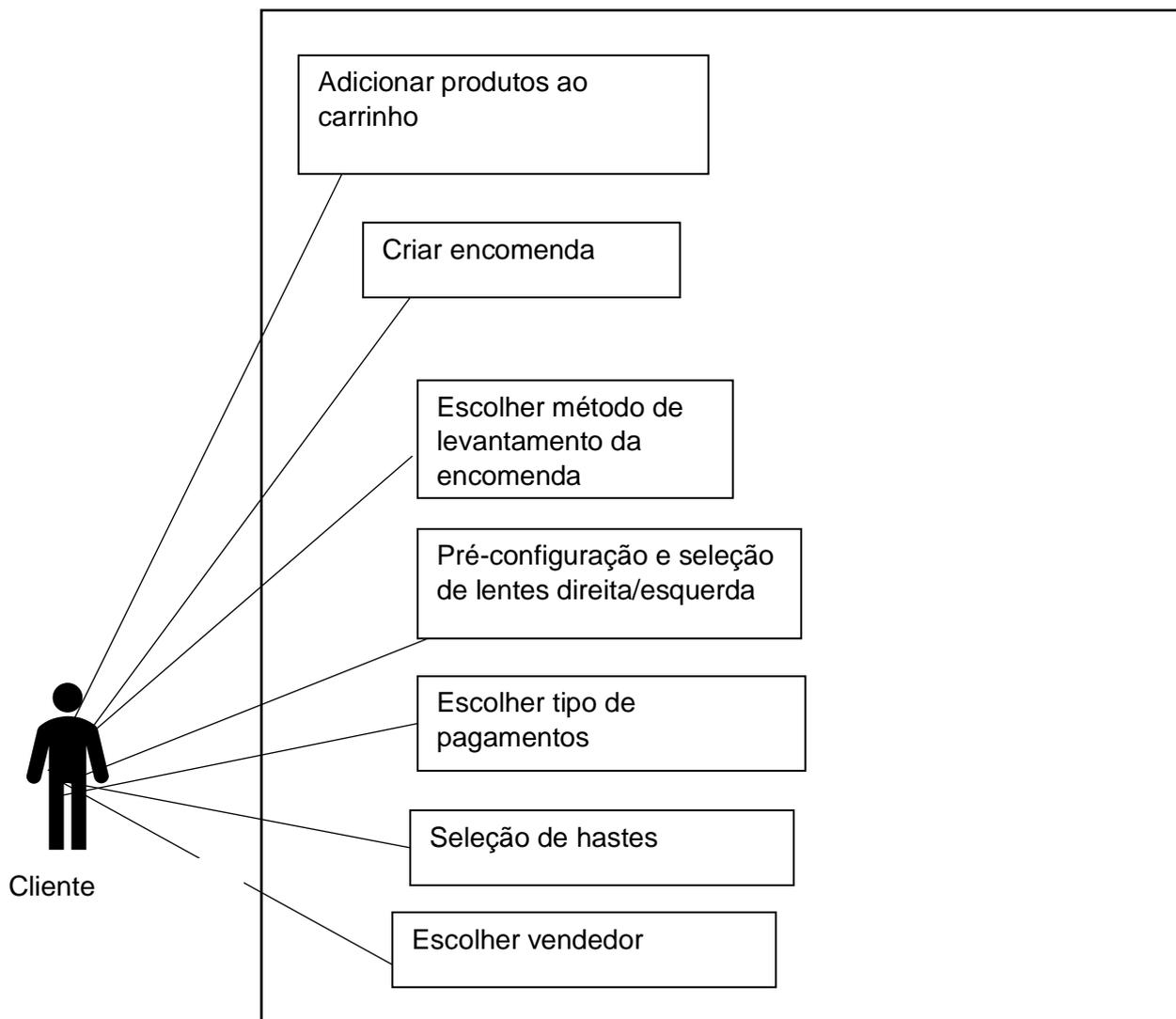


Diagrama de casos de uso da aplicação móvel

3.6- Softwares Usados na Aplicação LS Retail Loyalty

Para o desenvolvimento desta parte do trabalho foi usado o Visual Studio 2019. Para fazer as alterações necessárias na aplicação, o código original encontrava-se num repositório do qual se fez clone para o Visual Studio, o mesmo aconteceu para o projeto do *service*, onde se encontra as funções referentes a comunicação com o LS OMNI. Foi necessário também o uso do LS OMNI e do SQL Server para comunicar os dados com O BC, através do uso do WCF e o SQL Server para guardar dados temporários a comunicação da APP para o BC.

3.6.1-Visual Studio 2019

O Visual Studio é um ambiente de desenvolvimento integrado (IDE) da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# (C Sharp) e F# (F Sharp). Permite também o desenvolvimento na área da web, pois usa a plataforma do ASP.NET, para criar websites, aplicações web, serviços web e aplicações móveis (Visual Studio Code, 2020).

3.6.2-LS OMNI

O LS OMNI possui um conjunto de 3 componentes que são o serviço WCF em execução no IIS (Internet Information Service) para a comunicação de dados entre dois ambientes diferentes, a base de dados LS Omni SQL Server, onde possui várias tabelas que guardam alguns dados que serão enviados entre ambientes e o 3º componente pode ser uma aplicação móveis, soluções de comércio eletrónico, entre outros, apenas tem de comunicar por IIS.

O LS OMNI, possui uma componente importante com o envio de notificações e mensagens de correio eletrónico, ou seja, sempre que o estado da encomenda é alterado o cliente vai ser notificado, e assim permite um melhor acompanhamento da mesma.

Como podemos ver na figura 37, apresenta o funcionamento do LS OMNI, entre dois ambientes. Onde também é apresentado o WCF na comunicação entre os dois ambientes.

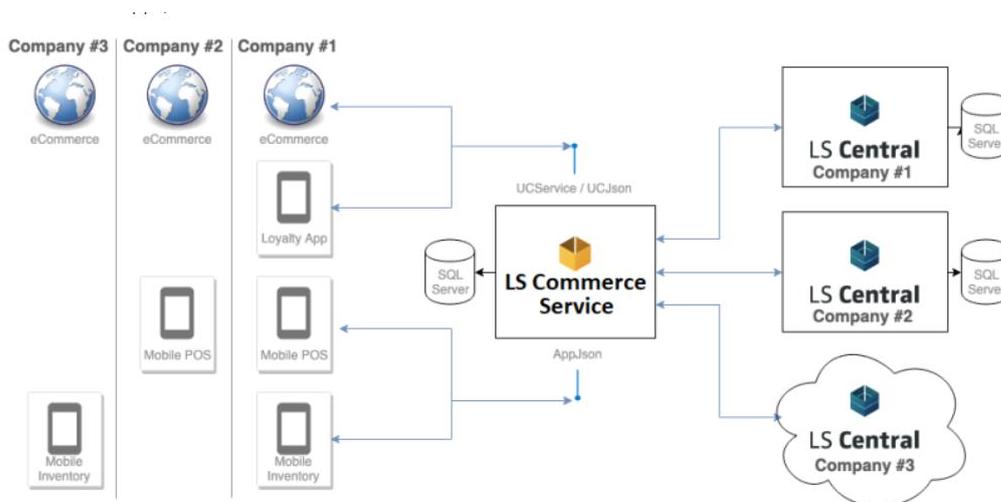


Figura 38: Apresentação do LS OMNI (Retail L. , 2021)

3.6.3-WCF

O WCF está ligado ao LSOmniService e permite fazer a comunicação de dados entre a aplicação móvel e o Business Central, este é uma parte da .Net Framework que permite construir aplicações distribuídas e orientadas a serviços (SOA). Esta tecnologia surgiu com o objetivo de unificar as até então existentes tecnologias de programação distribuídas. O cliente WCF usa *Simple Object Access Protocol* (SOAP) para efetuar a comunicação com os servidores, sendo que cliente e servidor são independentes do sistema operativo, hardware e plataforma de programação bem como a comunicação. Este mecanismo permite aos clientes invocar e consumir vários serviços, e um único serviço pode ser consumido por vários clientes, como podemos ver na figura 37

O WCF está ligado ao LSOmniService e permite fazer a comunicação de dados entre a aplicação móvel e o Business Central.

Algumas vantagens do WCF:

- O WCF pode ser configurado para funcionar independentemente do SOAP e usar o RSS (*Really Simple Syndication*).
- O WCF é uma das tecnologias de comunicação mais rápidas e oferece excelente desempenho em comparação com outras tecnologias da Microsoft.

- Para melhorar a comunicação, a velocidade de transmissão precisa ser otimizada. Isto é conseguido transmitindo dados XML com código binário em vez de texto sem formatação para diminuir a latência.
- A gestão do ciclo de vida do objeto e a gestão de transações distribuídas são aplicáveis a qualquer aplicação desenvolvida usando o WCF.
- O WCF usa as especificações do WS (*Web Services*) para fornecer confiabilidade, segurança e gerenciamento de transações.
- As mensagens podem ser colocadas em fila de espera, isto é uma vantagem pois não ocorrem atrasos, mesmo sob condições de tráfego intenso.
- O WCF foi projetado para comunicar com outros sistemas que não são do WCF, além dos vários sucessores e predecessores da tecnologia da Microsoft (What Is Windows Communication Foundation, 2020).

3.7- Implementação da Aplicação

Para desenvolver a aplicação relativamente à parte das óticas, foi necessário retirar de um repositório dois projetos diferentes. Um deles era referente ao desenvolvimento da aplicação e outro era referente à comunicação com o servidor. Os dois comunicavam entre si através das DLL's que LS Omni possuía. De seguida foi necessário fazer um estudo profundo ao comportamento da aplicação em geral, ou seja, foi preciso passar algum tempo a fazer *debug* na aplicação para entender de que forma é que esta tinha sido contruída e para entender as funcionalidades de cada função, só assim é que foi possível perceber o seu funcionamento e conseguiu-se passar a fase do desenvolvimento dos novos ecrãs relativos aos novos requisitos.

O planeamento desta nova parte da aplicação passou pela subdivisão dos passos a tomar desde que o cliente tem a receita dada pelo médico até que procede ao pagamento dos seus óculos *online* através da aplicação. Para tal, foi necessária uma estruturação pensada de acordo com a base de dados.

A partir das configurações existentes na base de dados e, atendendo as necessidades do utilizador foi ponderado, idealizado e desenvolvidos oito (8) novos *layouts* para assim proporcionar ao utilizador um compra mais rápida e sem ter a necessidade de sair de casa dos seus óculos. Cada um destes novos *layouts* conta com uma classe *activity* e *fragment* associada para permitir o seu funcionamento, existem outras classes, como por exemplo, onde são definidos os atributos de cada ecrã e outras que possuem funções que comunicam com as DLL's através dos *web services*,

permitindo a comunicação com a base de dados nos dois sentidos, tanto a trazer dados para a APP, como a levar da APP para a base de dados.

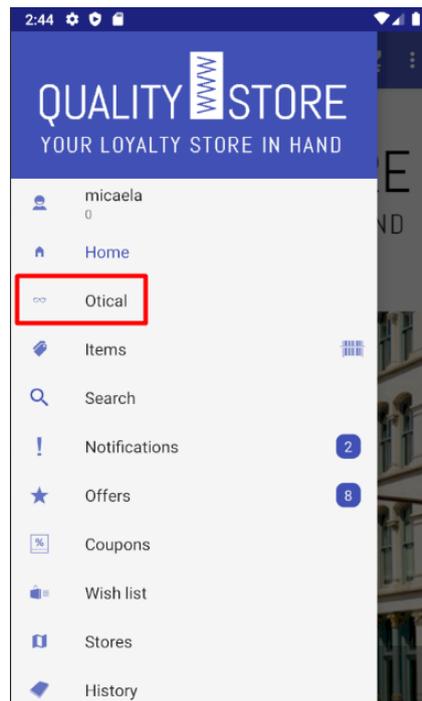


Figura 39: Menu principal com link para o ecrã da ótica

O ecrã da figura 38 pertencia à aplicação inicial, à qual apenas foi acrescentado o link para o ecrã inicial da ótica.

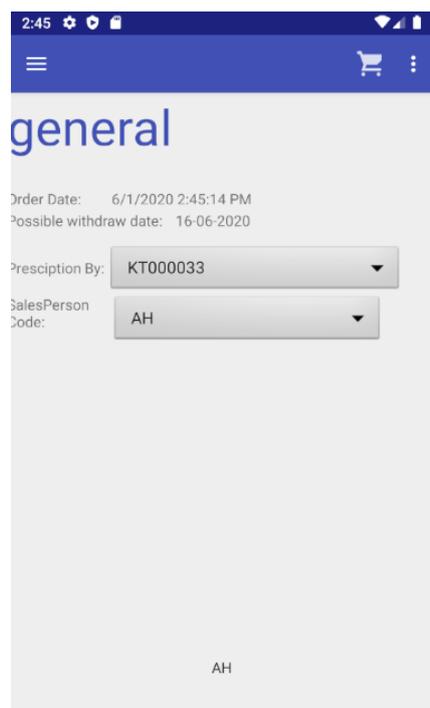


Figura 40: Ecrã inicial

Na figura 39 é apresentado ao utilizador a data do registo da encomenda e a data do possível levantamento. O utilizador pode também escolher quem fez a prescrição dos óculos.

The screenshot shows a mobile application interface with a blue header bar containing the text 'ARQUI' and a hamburger menu icon. Below the header, the word 'glass' is displayed in a large, blue, sans-serif font. The main content area features several dropdown menus for selection:

- Glass Type: BIFOCAL
- Chose Lens Material Type: POLYCARBON
- Glass / CL-Vendor: CZ
- Refraction by: AH
- Refractionist: Employee
- Order Type: CLAIM-GW

At the bottom of the form, there is a blue button labeled 'Next' and the text 'CLAIM-GW' centered below it.

Figura 41: Ecrã com informação dos óculos

Na figura 40 é possível escolher o tipo de lentes e o material das hastes, entre outras coisas.

ARQUI

left lens

Sphere: 2

Cylinder: 0

Axis: 0

Addition: 0

PD: 0

Base: Top

Segment Height: 0

Axis: 0

Prism: 0

Base Value: 0

Next

Figura 42: Ecrã para escolher a rarefação das lentes da esquerda

ARQUI

right lens

Sphere: 2

Cylinder: 2

Axis: 2

Addition: 2

PD: 2

Segment Height: 2

Axis: 2

Prism: 2

Base: Top

Base Value: 2

Next

Figura 43: Ecrã para escolher a rarefação das lentes da direita

Nos ecrãs da figura 41 e 42 a graduação das lentes pode ser escolhida consoante a receita médica, dependendo do tipo de doença que a pessoa tenha.

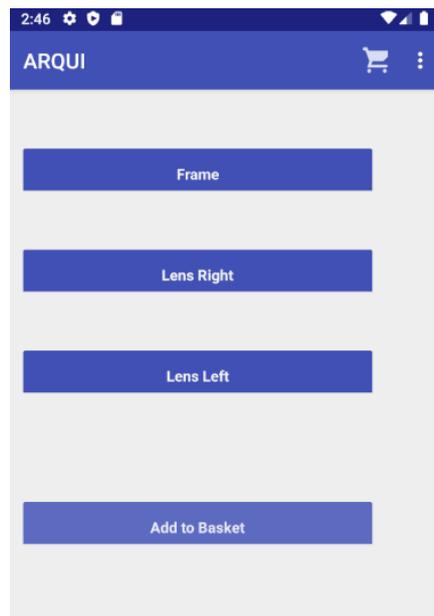


Figura 44: Ecrã para seleccionar as hastes e as lentes e para adicionar os produtos ao carrinho

A figura 43 mostra 4 botões disponíveis ao utilizador. O botão das *frames* permite a seleção entre todos os itens da base de dados e traz apenas os que possuem a categoria das *frames*. Os botões das lentes fazem o mesmo que este, mas trazem os itens com a categoria de lentes. Existem dois para a pessoa escolher a lente esquerda e lente direita.

O botão Add Basket permite adicionar os 3 produtos ao carrinho de compras quando estes já tiverem sido seleccionados, para que assim se possa finalizar a encomenda.

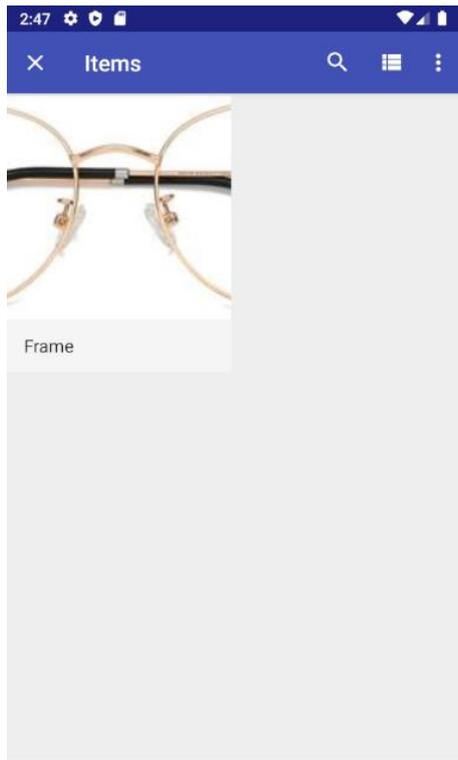


Figura 45: Seleção dos itens por frame

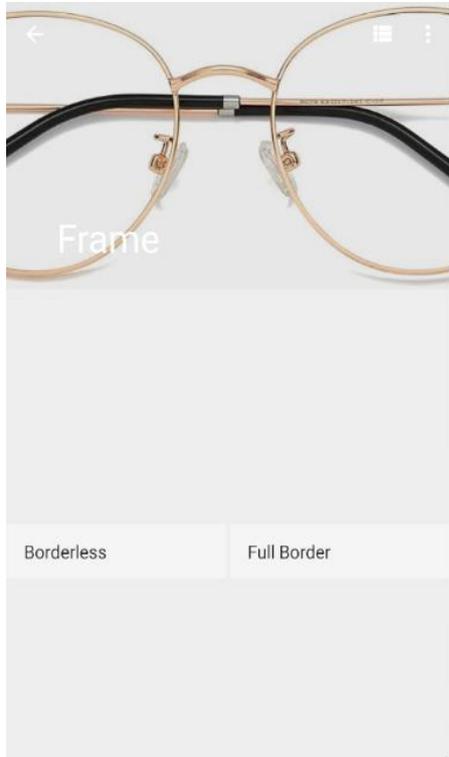


Figura 46: Categoria das Frames

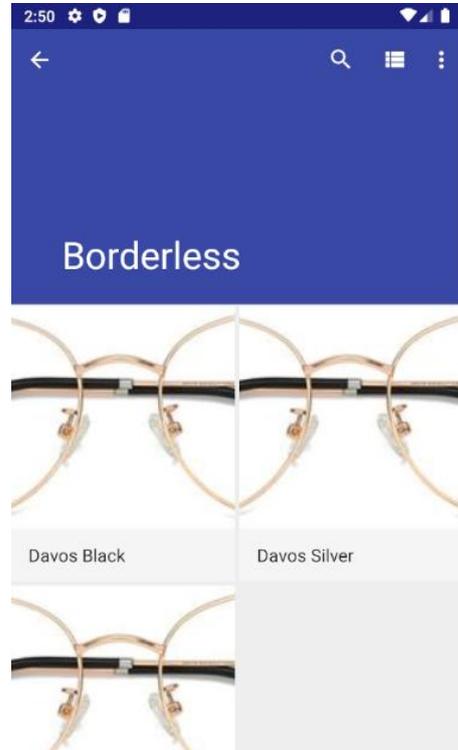


Figura 47: Sub-Categorias das Frames

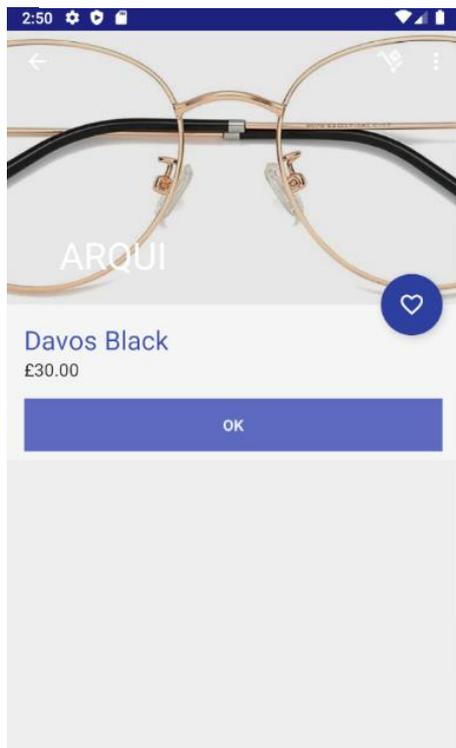


Figura 48: Adicionar um item à encomenda

Os ecrãs ilustrados nas figuras 44, 45, 46 e 47 são os ecrãs que seguem a sequência do botão das *frames* do ecrã 6. O mesmo acontece para o botão das lentes. Os ecrãs 44,45 e 46, são cópia dos ecrãs que já existentes na aplicação, quando selecionamos os produtos. O ecrã 47, já existia na aplicação, mas sofreu uma pequena alteração. O botão de adicionar ao carrinho, foi alterado para um botão de adicionar apenas a uma lista de itens, para que assim a encomenda seja adicionada toda ao carrinho, quando é clicado no botão do Add Basket e a lista tenha 3 produtos, ao mesmo tempo, para que não haja interferência de outros produtos que não sejam lentes nem *frame* na compra. Sempre que estes 3 produtos são adicionados ao carrinho, é efetuada uma verificação deste, para garantir que não há outros produtos que não pertençam as óticas e, caso haja, estes são eliminados.

A aplicação do LS Retail Loyalty, de uma forma geral permite fazer compras de diversos produtos, como pão, legumes, roupa, entre outros, para poder fazer compras é necessário estar logado na aplicação. Depois de adicionar produtos ao carrinho de compras, é permitido escolher se quer levantar em loja ou ser os produtos serem enviados para casa. É possível também escolher o método de pagamento.

De uma visão geral sobre a versão das Óticas desta aplicação o utilizador poderá encomendar os seus óculos com base nas especificações detalhadas na sua receita médica. Através dos dados na sua prescrição o utilizador poderá inserir na aplicação os campos detalhados da sua necessidade como por exemplo os axys, sphere, o cylinder entre outros. É proporcionado ao cliente a possibilidade de alteração caso algum dos campos seja mal preenchido em relação a graduação da lente. Posteriormente o cliente tem acesso à escolha do tipo de armação que deseja e também ao tipo de lente que pretende colocar nos seus óculos. Assim que tudo esteja de acordo com o utilizador a compra poderá ser finalizada.

3.8 - Dificuldades de Implementação

Durante a elaboração deste upgrade da aplicação deparei-me com várias dificuldades, nomeadamente uma em particular, sendo ela a compreensão do funcionamento da mesma. Para tal, procedi campo a campo a efetuação de *debug* diferenciado para cada passo tomado pelo utilizador da aplicação desde o seu *login* até *check-out* com as compras efetuadas.

Com este tipo de procedimento, foi-me possível perceber quais os aspetos mais importantes para a criação de novos *layouts* que iriam servir de interface gráfica para o utilizador, sendo que este foi o passo inicial durante todo este processo. Após uma melhor interiorização da aplicação foram concebidos oito *layouts* novos relacionados com as Óticas. Foi também necessário criar novas classes para ter acesso a dados de acordo com a base de dados, ou seja, ter acesso a dados de novas tabelas.

Este trabalho tem por base dois projetos de Visual Studio em que um deles possui a parte da construção da aplicação e o outro a comunicação com o servidor.

Depois de ter passado algum tempo a fazer *debug* na aplicação, comecei a criar as classes e deparei-me com alguns problemas como a passagem de dados de tabelas para as *dropdown list*. Tentei durante algum tempo resolver este problema, mas não consegui e então optei por usar listas estáticas, o que não é a melhor solução para o futuro, pois se houver alterações na base de dados, as listas da aplicação têm de ser alteradas também. Outro problema que tive foi com o carrinho de compras, pois a encomenda dos óculos tinha de ser feita sozinha sem outros produtos para isso tive de criar novos ecrãs em parte foram cópia dos que já existiam para poder adicionar os 3 produtos ao carrinho sozinhos, para que assim pudesse ser feita a encomenda da mesma forma que é feita do lado do Business Central. Teve de ser criado um botão para cada lente para podermos saber qual é que é a lente da direita e da esquerda.

Outro dos grandes problemas que tive foi ao finalizar encomenda, não estava a conseguir fazer isso pois o xmlport original, não tinha todos os campos necessários para levar a nova informação. Depois de algum tempo a tentar perceber o que era preciso fazer, percebi que tinha de alterar o projeto do servidor para levar também os novos campos e tive igualmente de acrescentar campos em tabelas e no xmlport respetivo do lado do Business Central. Isto teve de ser feito para atualizar as DLLs do LS Omni, pois é com este programa que é feita a comunicação de dados entre os dois ambientes. Para que a encomenda conseguisse ser criada do lado do Business Central, tive de alterar a minha base de dados para uma versão mais recente, para a versão BC140, pois esta possui as *codeunits* necessárias para a passagem de dados o que não acontecia na versão BC130.

Tive também de criar eventos do lado do projeto do AL, para poder inserir os dados dos campos da extensão nas tabelas Sales Header e a Sales Line, através dos dados que foram inseridos nas Customer Order Header e na Customer Order Line.

Para atualizar as DLL's do LS Omini, é necessário correr a instalação do LS Omni, através da pasta do projeto do Service, assim as DLL's ficam atualizadas consoante as alterações do que se encontram no projeto.

Quando se altera os XmlPort's do lado do BC, é necessário efetuar no projeto do servidor um *update* das *webreferences*, para que este consiga reconhecer as atualizações feitas e possamos chamar os campos novos que foram adicionados.

3.9-Testes de Software Realizados no Desenvolvimento das Duas Componentes

Devido à falta de tempo, apenas foram realizados testes laboratoriais ao longo do desenvolvimento de cada uma das partes, para corrigir erros que iam aparecendo e verificar se tudo estava a funcionar. No final, foram efetuados testes para verificar se todos os requisitos eram cumpridos e se o resultado final estava de acordo com o pretendido inicialmente.

No caso da migração, foi testado a criação de clientes (exemplo figura 48), a criação da *glass sales order* e a sua respetiva configuração (exemplo figura 49 e 50), a mudança do seu estado, como podemos ver na figura 51 e a impressão do *report* com a respetiva fatura, como exemplo apresentado na figura 52.

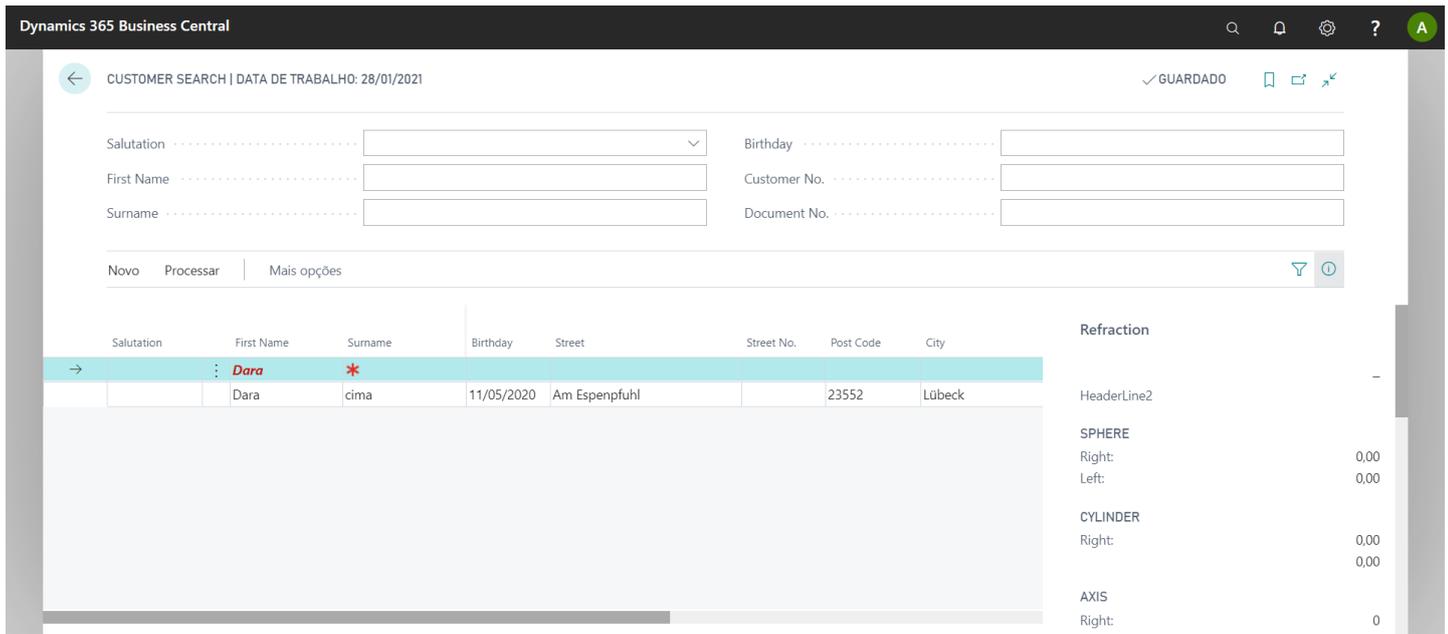


Figura 49: Criação de clientes na migração

Refraction		Compare Measuring	
RIGHT		LEFT	
Sphere	<input type="text" value="0,00"/>	<input type="text" value="0,00"/>	<input type="text" value="0,00"/>
Cylinder	<input type="text" value="0,00"/>	<input type="text" value="0,00"/>	<input type="text" value="0,00"/>
Axis	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Addition	<input type="text" value="0,00"/>	<input type="text" value="0,00"/>	<input type="text" value="0,00"/>
PD	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Segment Height	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Visus	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Prism	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Base	<input type="text" value="v"/>	<input type="text" value="v"/>	<input type="text" value="v"/>
Base Value	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Figura 50: Escolha do índice de refração das lentes

Do lado da aplicação, é criado o formulário em diferentes *layouts* seguindo a mesma estrutura da página da extensão, *glass sales order*, e depois de criar a encomenda, é feita a verificação para garantir que foi criada no lado do BC, e se é possível passar a fatura e fazer o respetivo pagamento, quando se consegue isso os testes correm bem.

Relativamente aos testes de sistema, foi testada a comunicação entre os diferentes sistemas.

No caso da migração foi testada a comunicação do AL, com o BC e com a base de dados.

No lado da APP, a comunicação foi feita entre 3 sistemas, que são a APP, o LS OMNI, e o BC e deve-se garantir que esta tem ligação há internet. Esta comunicação tem de ser garantida para que as encomendas possam ser feitas e os clientes fiquem satisfeitos. Para verificar a comunicação com o LS OMNI é feito o ping, através do url <http://localhost/LSOmniService/appjson.svc/ping>, no browser ou na APP, sendo que na APP tem de ser sempre feito sempre que houver uma alteração para esta atualizar (ver figura 53). Se obtivermos o Pong OK (figura 54) então a comunicação com o LS Omni está a funcionar e é então possível enviar dados para o Business Central, através de webservices que estão na Codeunit do OmniWapper (figura 55).

FRAME		RIGHT	LEFT
Item No.	500001	F-R-A2	F-R-A1
Description	Recife black
Diameter	
		Color	...
Amount	41,65	175,00	125,00
Demand Cove...	Store	SalesLineRigh... Vendor	SalesLineLeft... Vendor
Status		SalesLineRigh...	SalesLineLeft...

Figura 51: Escolha dos produtos das lentes e das hastes

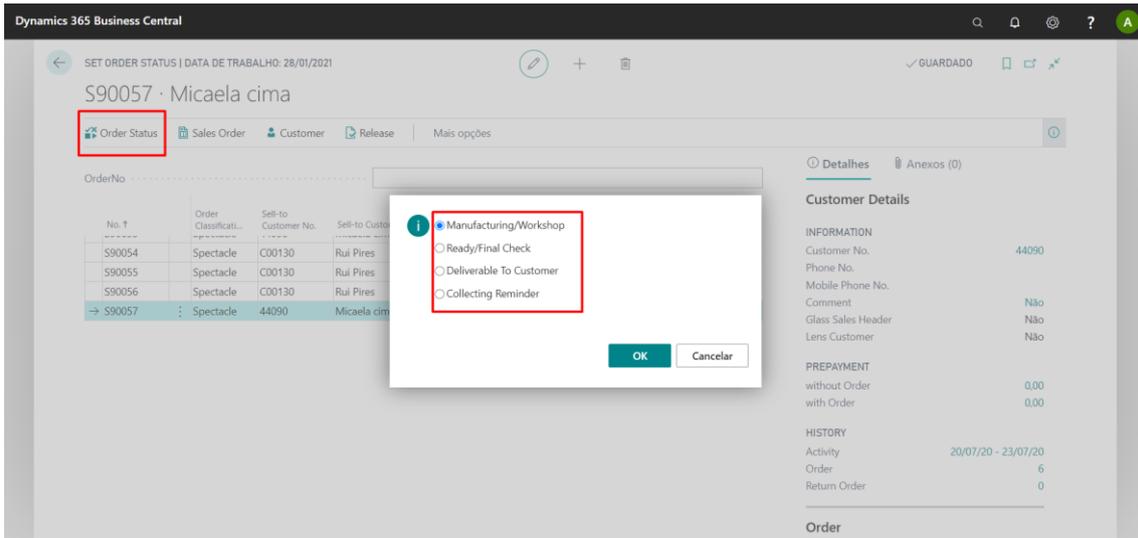


Figura 52: Alteração do estado das glass sales orders

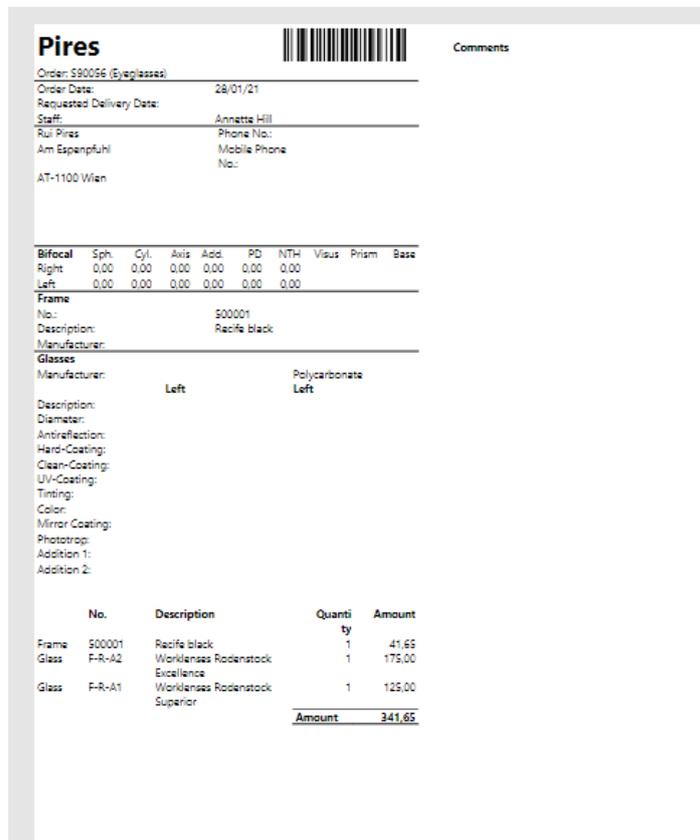


Figura 53: Exemplo da fatura do cliente

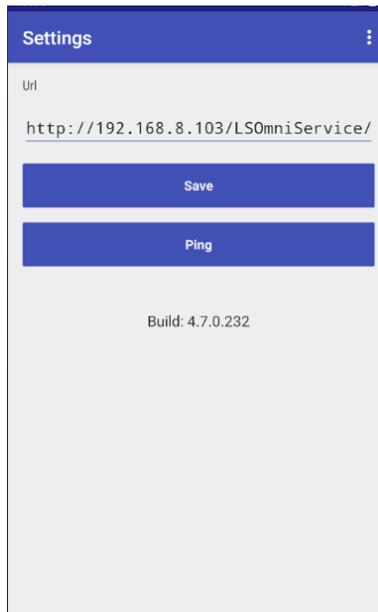


Figura 54: Ecrã para fazer ping na app

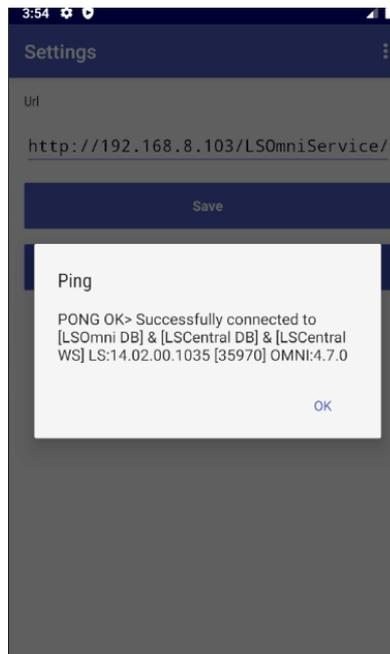


Figura 55: Resposta ao Ping

Web Services - Microsoft Dynamics NAV client - connected to Dynamics 365 Business Central

CRONUS LS 1402 W1 Demo | Departments | Administration | IT Administration | Services | Web Services

HOME ACTIONS

Web Services -

Object Type	Object ID	Object Name	Service Name	All Tena.	Publ.	OData V4 URL	OData URL	SOAP URL
Codeunit	10033143	MobilePosSave	MobilePosSave	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033024	MobilePosSetDeviceID	MobilePosSetDeviceID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033055	MobilePosSuspend	MobilePosSuspend	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033031	MobilePosZReport	MobilePosZReport	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033100	OmniWrapper	OmniWrapper	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033125	ProcessTransMemberPoints	ProcessTransMemberPoints	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	99009614	RetailMessageGetActiveList	RetailMessageGetActiveList	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033129	RetailMessageRead	RetailMessageRead	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033131	RetailMessageUpdRcpStatus	RetailMessageUpdRcpStatus	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10012735	POS Web API	RetailWebAPI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	99009510	Web Services	RetailWebServices	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033075	SendDataEntry	SendDataEntry	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033085	SendDocumentImage	SendDocumentImage	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033103	SendMemberEmail	SendMemberEmail	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033109	SendMemberProcessEntry	SendMemberProcessEntry	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033105	SendPosStartStatus	SendPosStartStatus	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033107	SendPosTransBackup	SendPosTransBackup	<input type="checkbox"/>	<input checked="" type="checkbox"/>	http://ARQUI03-MDC.Arquiconsult.com:1447/BC140/WS/CRONUS%20LS%201402%20W1%20Demo/Codeunit/SendPosStartStatus	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033111	SendPosTransSusp	SendPosTransSusp	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033163	SendSerialCoupon	SendSerialCoupon	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033071	SendStaff	SendStaff	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033045	SendTransaction	SendTransaction	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...
Codeunit	10033049	SendTransactionHeader	SendTransactionHeader	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Not applicable	Not applicable	http://ARQUI03-MDC.Arquiconsult.com:1447/BC14...

CRONUS LS 1402 W1 Demo | 21 de maio de 2019 | ARQUICONSULT/MICAEIA/CMIA

Figura 56: Codeunit do OmniWrapper

Do lado do BC é sempre necessário testar a conexão com os web services através dos *test connection*. Se a resposta for positiva os webservices estão prontos a ser usados, como podemos ver na figura 55 e 56.

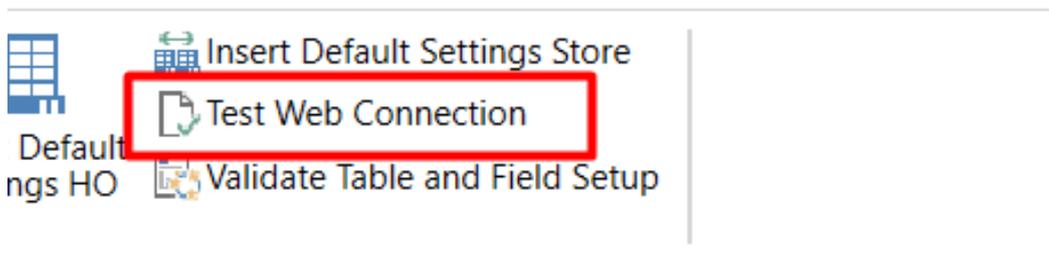


Figura 57: Página para fazer o test connection

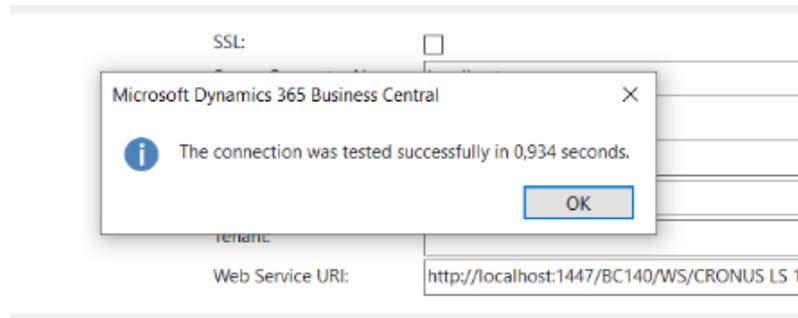


Figura 58: Resposta ao test connection

Por falta de tempo não houve testes do lado do cliente e apenas foi efetuada a simulação do que é suposto acontecer nesse lado para verificar se tudo estava a funcionar.

4 - Conclusões

Este relatório pretende relatar as etapas principais do estágio ocorrido na empresa Arquiconsult, sediada em Vila Real.

O trabalho de estágio teve como principal objetivo a implementação da extensão, numa nova versão, a criação da importação no formato SF6 e por fim a adição de novas funcionalidades à aplicação móvel. Estas alterações surgiram com base na necessidade de uma atualização nos campos fulcrais para o desenvolvimento deste trabalho, para que a utilização da aplicação se tornasse mais completa e facilita-se a migração para versões futuras. Posto isto, o propósito foi adicionar à base de dados *standard*, novos componentes específicos para uma loja de óculos, em todas as partes desenvolvidas. Para tal, numa fase inicial foi necessária fazer uma recolha de informação sobre o LS RETAIL, bem como a sua vertente ligada às Óticas. Dentro destas dificuldades foi também pertinente uma pesquisa mais específica relacionada com pequenos detalhes necessários para o desenvolvimento deste trabalho, tais como, ERP, Business Central, Testes de Software.

Para a atingir de forma aprimorada o objetivo foi necessário obter informação sobre o LS RETAIL. Como descrito no capítulo 1 foi pertinente identificar, as marcas (clientes) em que este é usado atualmente e identificar as principais dificuldades e vantagens do seu uso.

O capítulo 2 realça a necessidade de assimilar conhecimentos sobre ERP, vantagens e algumas desvantagens da sua utilização e alguns exemplos. É apresentada também alguma informação sobre o LS RETAIL ligado às óticas. É feita uma descrição geral sobre o Business Central, mostrando diversos exemplos de objetos. Neste capítulo foi também apresentado informação sobre os testes de *software*, de que forma é que devem ser realizados e quais os diferentes tipos de testes.

Para a elaboração das tarefas da migração do *add-on* proposta inicialmente, no plano de trabalhos, foi necessário definir quais os requisitos funcionais e casos de uso, diferentes soluções de *software* a serem usadas. De seguida ocorreu a implementação da migração, como podemos ver pela análise do subcapítulo 3.1 e 3.3. Ao longo do desenvolvimento surgiram diversos problemas, que foram devidamente apresentados no subcapítulo 3.2. Para a implementação do SF6, foi necessário fazer uma análise das normas e de seguida, fez-se a respetiva implementação esta encontra-se no subcapítulo 3.4. Relativamente à adição de novas funcionalidades na aplicação móvel, inicialmente definiram-se os requisitos, os respetivos casos de uso e identificaram-se os produtos de *software* a serem usados. Por último fez-se um estudo da aplicação existente que levou

algum tempo, para depois prosseguir para a implementação, podemos ver a descrição desta componente no subcapítulo 3.7 e 3.8.

Ao longo do desenvolvimento das diferentes tarefas foram realizados testes para corrigir erros existentes e verificar se as funcionalidades estavam de acordo com os requisitos, esta descrição encontra-se no subcapítulo 3.9. Os resultados atingidos cumpriram o objetivo, ou seja, conseguiu-se registar novos clientes, criar a ordem de vendas dos óculos e passar a respetiva fatura, para os mesmos. No caso da aplicação móvel os testes apenas foram realizados ao longo da implementação e no final, para confirmar que tudo funcionava. Conclui-se que era possível criar a encomenda do lado da aplicação e finalizar a mesma do lado do BC, seguindo os mesmos passos da ordem de venda dos óculos.

Pela análise feita anteriormente podemos concluir que as tarefas definidas inicialmente no plano de trabalhos foram realizadas, estando a descrição destas ao longo do capítulo 3. A descrição da migração é apresentada no subcapítulo 3.1 ao 3.3, o SF6 encontra-se descrito no subcapítulo 3.4. O desenvolvimento da aplicação móvel, é descrito no subcapítulo 3.5 ao 3.8. Já os testes são apresentados no subcapítulo 3.9.

A parte dos testes ficou aquém do que era inicialmente pretendido pois apenas foram feitos testes ao longo da criação dos diferentes componentes devido à falta de tempo. A tarefa da migração inicialmente foi definida como sendo realizada em 1 mês, mas acabou por demorar mais tempo, devido aos problemas que apareceram ao longo do desenvolvimento. No caso da aplicação móvel, esta também demorou mais tempo que o pretendido, pois teve de ser realizado inicialmente um conhecimento integral da app já existente, para depois iniciar o desenvolvimento.

Houve alguns problemas na parte da migração como foram referidos no subcapítulo 3.2.1. Na parte da aplicação esta também teve alguns problemas como foram apresentados no subcapítulo 3.8.

Para trabalhos futuros, serão propostos a realização de testes mais profundos, uma nova migração para versões mais recentes do BC e a integração com o POS, para finalizar a venda do produto em loja.

No caso da implementação do SF6, será proposto a para realização no futuro a alteração para as novas normas.

Sobre a aplicação móvel, esta deve ser melhorada, para passar a ser completamente dinâmica em relação aos dados que devem ser recebidos da base de dados, deverá também comunicar com versões do BC, mais recentes. Ao nível de testes esta também deverá passar pelo processo de testes mais profundos por equipas especializadas.

Bibliografia

- Alecrim, E. (2020). *O que é ERP e para que serve?* Obtido de InfoWester: <https://www.infowester.com/erp.php>. Acesso em: 23 ago. 2020
- Andersin, J. (2004). *TPI – a model for Test Process Improvement*. Obtido de Seminar on Quality Models for Software Engineering: <https://www.cs.helsinki.fi/u/paakki/Andersin.pdf>. Acesso em: 23 ago. 2020
- Are you getting the most out of your loyalty program?* (2020). Obtido de LS Retail: <https://www.lsretail.com/blog/are-you-getting-the-most-out-of-your-loyalty>. Acesso em: 23 ago. 2020
- Arquiconsult*. (2020). Obtido de Arquiconsult: <https://www.arquiconsult.com/pt/>. Acesso em: 23 ago. 2020
- Asserbiz. (25 de Julho de 2016). *APLICAÇÕES MÓVEIS PARA EMPRESAS: VALE A PENA PARA O SI? O QUE DEVE CONSIDERAR?* Obtido de Asserbiz: <https://asserbiz.pt/aplicacoes-moveis-para-empresas-vale-a-pena-para-o-seu-o-que-deve-considerar/>. Acesso em: 23 ago. 2020
- B. Kitchenham and S. L. Pfleeger. (Jan. 1996). "Software quality: the elusive target [special issues section],"
- Barrett, S. (23 de Abril de 2018). *HOW IOT ENHANCES ERP*. Obtido de ANEGIS: <https://www.anegis.com/articles/how-iot-enhances-erp>. Acesso em: 23 ago. 2020
- Bitterncourt, P. L. (2020). *MC714, Sistemas Distribuidos*. Obtido de <https://www.ic.unicamp.br/~bit/mc714/aulas/aula12.pdf>. Acesso em: 23 ago. 2020
- Brodie, M. (1 de june de 2017). *Installing NAV 2015 Build 41779 using a 3 tier setup. Error occurs "Cannot establish a connection to the SQL Server/Database."*. Obtido de Capricorn Dynamics.
- Burnstein, S. &. (1999). A testing maturity model for software test process assessment and improvement.
- Business Central*. (2020). Obtido de Microsoft: <https://dynamics.microsoft.com/pt-pt/business-central/overview/>
- Caetano, C. (s.d.). *Gestão de Testes Ferramentaas Open source e melhores práticas na gestão de testes*. Engenharia de software Magazine.
- Chopra, S. e. (2003). Gerenciamento da cadeia de suprimentos: estratégia, planejamento, e operação. Em *Gerenciamento da cadeia de suprimentos: estratégia, planejamento, e operação* (pp. 465-465).
- Faria, T. e. (3 de 4 de 2021). *O que é um sistema POS?* Obtido de Controlex Sistemas: <https://controlexsistemas.com/sistema-pos-o-que-e-vantagens/>. Acesso em: 23 ago. 2020

- Ferreirinha, M. S. (2017). *O papel das aplicações móveis no turismo – o caso Zarco*. PORTO: INSTITUTO SUPERIOR DE CONTABILIDADE E ADMINISTRAÇÃO DO PORTO.
- File Specification Lens Product Catalogue*. (s.d.). Obtido de B2Boptic: <http://files.b2boptic.com/sf6/oldversion/ProKatGla6.7.4-english.pdf>. Acesso em: 23 ago. 2020
- Fonseca, L. (4 de 4 de 2021). *Estratégia Omnichannel: uma nova experiência de consumo*. Obtido de rockcontent: <https://rockcontent.com/br/blog/omnichannel/>. Acesso em: 23 ago. 2020
- HashMicro. (2020). *6 Major Challenges In Retail Industry*. Obtido de BusinessTech ERP and Business Management Blog: <https://www.hashmicro.com/blog/6-major-challenges-in-retail-industry/>. Acesso em: 23 ago. 2020
- ISO. (2019). *The ISO/IEC 25000 series of standards*. Obtido de Iso25000: <https://iso25000.com/index.php/en/iso-25000-standards>. Acesso em: 23 ago. 2020
- LS Retail*. (2020). Obtido de Ls Retail: <https://www.lsretail.com/>. Acesso em: 23 ago. 2020
- LSRetail*. (05 de Maio de 2020). Obtido de LSRetail: <https://portal.lsretail.com/>. Acesso em: 23 ago. 2020
- Microsoft. (4 de 4 de 2021). *Microsoft*. Obtido de Programming in C/AL: <https://docs.microsoft.com/en-us/dynamics-nav/programming-in-c-al>. Acesso em: 23 ago. 2020
- Microsoft. (4 de 4 de 2021). *Programming in AL*. Obtido de Microsoft: <https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-programming-in-al>. Acesso em: 23 ago. 2020
- Microsoft SQL Server*. (2020). Obtido de Wikipedia: https://pt.wikipedia.org/wiki/Microsoft_SQL_Server. Acesso em: 23 ago. 2020
- Miguel Mazedo, L. T. (2020). Aplicações móveis para o turismo cultural: Caminhos de Santiago. *Revista Lusófona de Estudos Culturais*, 165-184.
- Miranda, N. (2017). *Desenvolvimento de Aplicações Móveis recorrendo a frameworks*. Coimbra: Universidade de Coimbra.
- Moura, B. C. (2006). *Logística*. Lisboa: Editora Centro Atlântico.
- Naik, K. a. (2011). *Software testing and quality assurance: theory and practice*. John Wiley & Sons.
- ODATA. (4 de 4 de 2021). *ODATA*. Obtido de ODATA: <https://www.odata.org/>
- Padilha, T. C., & Marins, F. A. (2005). *Sistemas ERP: características, custos e tendências*. Obtido de Scielo: <https://www.totvs.com/blog/erp/o-que-e-erp/>
- Paulk, M. (2002). Capability maturity model for software. *Encyclopedia of Software Engineering*.
- Planet, N. (2020). *LS Retail Architecture With Picture*. Obtido de Navision Planet.

- Pressman, R. (s.d.). *Software Engineering: A Practitioner's Approach*. Em R. Pressman, *Software Engineering: A Practitioner's Approach*.
- PRO, N. (24 de OCTOBER de 2020). *¿Por qué implementar un Sistema de Gestión ERP en mi empresa?* Obtido de ntxPRO: <https://www.ntxpro.net/erp/por-que-implementar-un-sistema-de-gestion-erp/>. Acesso em: 23 ago. 2020
- Retail, O. f. (4 de 4 de 2021). *Optical for LS Retail*. Obtido de Optical for LS Retail: <https://www.arquiconsult.com/en/sectors/retail-commerce/optical-for-ls-retail.html>. Acesso em: 23 ago. 2020
- Santos, V. F. (2018 de Agosto de 26). *Os 15 principais sistemas de software ERP*. Obtido de BLOG MELHORIA DE PROCESSOS: <https://www.fm2s.com.br/os-15-principais-sistemas-de-software-erp/>. Acesso em: 23 ago. 2020
- Simkin, M. G. (2014). *Core concepts of accounting information systems*. John Wiley & Sons.
- Stair, R. (1998). *Princípios de Sistemas de Informação: uma Abordagem Gerencial*. 2.ed. Editora LTC.
- STAIR, R. M., & REYNOLDS, G. W. (2006). *Princípios de Sistemas de Informação*. 6ª ed. São Paulo: São Paulo: Cambridge.
- Stefani, A., & Xenos, M. (2008). E-commerce system quality assessment using a model based on ISO 9126 and Belief Networks. *Software Quality Journal*.
- Suwanassart, T., Burnstein, I., & Carlson, R. (1996). Developing a Testing Maturity Model for software test process evaluation and improvement. *Proceedings International Test Conference 1996. Test and Design Validity*.
- Técnicas, A. -A. (s.d.). *Engenharia de software - Requisitos e avaliação da qualidade de produto de software (SQuaRE) - Guia e modelo de referência para medição*. Obtido de <https://www.abntcatalogo.com.br/norma.aspx?ID=38836>. Acesso em: 23 ago. 2020
- Turban, E., Rainer, R. K., & Potter, R. E. (2005). *Administração de tecnologia da informação*. Elsevier.
- Visual Studio Code*. (2020). Obtido de Microsoft: <https://code.visualstudio.com/>. Acesso em: 23 ago. 2020
- Visual Studio Code*. (2020). Obtido de Wikipedia: https://pt.wikipedia.org/wiki/Visual_Studio_Code. Acesso em: 23 ago. 2020
- What is PowerShell?* (2020). Obtido de Microsoft: <https://docs.microsoft.com/pt-pt/powershell/scripting/overview?view=powershell-7>. Acesso em: 23 ago. 2020
- What Is Windows Communication Foundation*. (2020). Obtido de Microsoft: <https://docs.microsoft.com/en-us/dotnet/framework/wcf/whats-wcf>. Acesso em: 23 ago. 2020
- wiki.b2boptic. (4 de 4 de 2021). *wiki.b2boptic*. Obtido de wiki.b2boptic: <http://wiki.b2boptic.com/en:lenscatalog:index>. Acesso em: 23 ago. 2020