

Mestrado em Engenharia Informática  
Estágio  
Relatório Final

**SIGER**

Diogo José Santos Dias  
djdias@student.dei.uc.pt

Orientador:

Vasco Pereira

Data: 04 de Setembro de 2013



**FCTUC** DEPARTAMENTO  
**DE ENGENHARIA INFORMÁTICA**  
FACULDADE DE CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA

## RESUMO

O projecto SIGER (Sistema Integrado de Gestão e Exploração para Restauração) visa desenvolver um sistema de apoio ao negócio da restauração/retalho através da aplicação de novas tecnologias e *Hardware* especializado. O principal objectivo do produto é garantir que de forma modular, o sistema seja capaz de se adaptar às necessidades de cada utilizador mas ao mesmo tempo nunca perder a visão global do negócio.

No âmbito deste estágio irão ser desenvolvidos os módulos de Ponto de Venda (POS) e de controlo de *Drink-Dispensers*. O módulo de Ponto de Venda será a base do sistema SIGER e responsável por garantir as funcionalidades de registo de vendas e facturação. Já o módulo de *Drink-Dispensers* irá trazer algum factor de inovação em relação aos produtos concorrentes, permitindo realizar uma gestão mais eficaz e controlada das bebidas servidas dentro de um determinado estabelecimento.

## PALAVRAS-CHAVE

“Controlo de bebidas”, “Integração”, “NFC”, “Ponto de Venda”, “Restauração”, “RFID”

## ÍNDICE

Resumo.....	2
Palavras-Chave .....	2
Índice .....	3
Lista de Figuras .....	5
Lista de Abreviaturas e Acrónimos .....	6
1. Introdução.....	7
1.1. Enquadramento.....	7
1.2. Equipa.....	7
1.3. Metodologia de Desenvolvimento.....	8
1.4. Planeamento .....	10
2. Estado da Arte.....	14
2.1. Produtos concorrentes.....	14
2.1.1. WinRest FrontOffice PRO.....	14
2.1.2. ZsRest FrontOffice.....	15
2.1.3. Sage GesRest II .....	15
2.1.4. GestWin POS .....	15
2.1.5. HarborTouch.....	16
2.1.6. Quadro comparativo.....	16
2.2. Tecnologias Envolvidas.....	19
2.2.1. Sistemas de Controlo de Bebidas.....	19
2.2.2. RFID e NFC.....	19
2.2.3. Caso de estudo.....	21
2.2.4. Cloud vs Local .....	22
3. Projecto.....	24
3.1. Análise de Requisitos.....	24
3.2. Especificação Funcional .....	26
3.3. Arquitectura .....	28
3.3.1. Arquitectura Funcional.....	29
3.3.2. Arquitectura Técnica.....	31
3.4. Modelo de Dados .....	32

3.5.	Tecnologias .....	37
3.6.	Ferramentas e Bibliotecas de Apoio .....	38
3.7.	Testes de exploração Realizados .....	39
4.	Implementação .....	43
4.1.	Trabalho desenvolvido.....	43
4.2.	Desafios Encontrados.....	49
4.2.1.	Modelo de dados .....	49
4.2.2.	Estruturação do código .....	49
4.2.3.	Comunicação Esc/p (impressoras) .....	50
4.2.4.	Integração com o Master-Receiver.....	50
4.2.5.	Certificação da aplicação .....	51
4.2.6.	Testes.....	51
5.	Conclusão e Próximas Tarefas .....	53
5.1.	Conclusão.....	53
5.2.	Trabalho Futuro.....	53
	Referências.....	55
	Anexos.....	57

## LISTA DE FIGURAS

FIGURA 1 - CALENDARIZAÇÃO DO PLANO DE TRABALHOS [VERSÃO PLANEADA] .....	12
FIGURA 2 - CALENDARIZAÇÃO DO PLANO DE TRABALHOS [VERSÃO REAL] .....	13
FIGURA 3 - POSSIBILIDADES DE APLICAÇÃO DAS TECNOLOGIAS RFID/NFC.....	20
FIGURA 4 – ÁREA DE REGISTO (ESPECIFICAÇÃO INICIAL) .....	28
FIGURA 5 – ÁREA DE REGISTO (PRODUTO FINAL).....	28
FIGURA 6 - ARQUITECTURA FUNCIONAL.....	29
FIGURA 7 - ARQUITECTURA TÉCNICA.....	31
FIGURA 8 - CONJUNTO DE ENTIDADES E RELACIONAMENTOS PARA GESTÃO DE ITENS DE VENDA .....	33
FIGURA 9 - CONJUNTO DE ENTIDADES E RELACIONAMENTOS PARA GRUPOS E PERMISSÕES .....	34
FIGURA 10 - CONJUNTO DE ENTIDADES E RELACIONAMENTOS PARA GESTÃO DE SESSÕES .....	35
FIGURA 11 - CONJUNTO DE ENTIDADES E RELACIONAMENTO PARA GESTÃO DE DOCUMENTOS .....	36
FIGURA 12 - CONJUNTO DE ENTIDADES E RELACIONAMENTOS PARA CONTROLO DE DD .....	37
FIGURA 13 - TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO .....	38
FIGURA 14 - DISPOSITIVO <i>MASTER-RECEIVER</i> .....	39
FIGURA 15 - DISPOSITIVO <i>POUR SPOUT</i> .....	39
FIGURA 16 - DISPOSITIVO <i>OPTIC PEARL</i> .....	40
FIGURA 17 - SELECÇÃO DE PERFIS & LOGIN.....	43
FIGURA 18 - MENU CONFIGURAÇÃO .....	44
FIGURA 19 - REGISTO DE VENDAS .....	44
FIGURA 20 - FINALIZAR VENDA [PAGAMENTO] .....	44
FIGURA 21 - IMPRESSORA DE ECRÃ .....	45
FIGURA 22 - CONSULTA DE DOCUMENTOS.....	46
FIGURA 23 - EXPORTAÇÃO DE FICHEIROS SAF-T.....	47
FIGURA 24 - GESTÃO DE UTILIZADORES.....	48
FIGURA 25 - CONTROLO DE DDs.....	48
FIGURA 26 - INTERFACE PARA O CORE DA APLICAÇÃO .....	49

**LISTA DE ABREVIATURAS E ACRÓNIMOS**

<b>DD</b>	Drink-Dispenser
<b>MR</b>	Master-Receiver
<b>NFC</b>	Near Field Communication
<b>PDA</b>	Personal Digital Assistant
<b>POS</b>	Point of Sale
<b>RFID</b>	Radio Frequency Identification
<b>ESC/P</b>	Epson Standard Code for Printers

# 1. INTRODUÇÃO

Este documento visa apresentar em detalhe todo o processo de estudo, planeamento e desenvolvimento envolvido na realização do projecto SIGER (Sistema Integrado de Gestão e Exploração para a Restauração). O projecto está inserido no âmbito do estágio curricular do curso de Mestrado em Engenharia Informática, leccionado na Faculdade de Ciências e Tecnologias da Universidade de Coimbra.

## 1.1. ENQUADRAMENTO

O projecto em que o estagiário se encontra inserido faz parte de uma proposta inovadora apresentada pela empresa **Softkom2**. Esta empresa encontra-se neste momento em fase de incubação no Instituto Pedro Nunes.

Actualmente o negócio de venda de Artigos e/ou Serviços ao publico, de pequena ou média dimensão, depara-se com alguns problemas ou limitações que muitas das vezes comprometem seriamente a viabilidade deste tipo de negócios. Apesar de em grande parte das situações já ser possível encontrar negócios que recorrem ao uso de pontos de venda electrónicos para auxiliar na gestão do negócio ou simplesmente por necessidades legais, a verdade é que nem sempre esse tipo de soluções consegue oferecer as respostas que os utilizadores procuram. As razões podem ser:

- A solução encontrada é complexa demais para o cenário de utilização pretendido e a falta de flexibilidade faz com que muitas das vezes o cliente seja obrigado a pagar por funcionalidades que nunca irá sequer utilizar.
- A solução encontrada não é capaz de oferecer uma visão global do negócio visto que não existe integração entre as diferentes aplicações usadas (ex.: facturação, gestão de stocks, controlo de entradas/saídas no estabelecimento).
- Falta controlo sobre as operações realizadas pelos empregados, o que permite que existam actividades paralelas dentro do negócio e desvios muitas das vezes indetectáveis.
- Dificuldade na realização de inventários e gestão de *stocks*, devido à sua morosidade.
- As soluções propostas muitas das vezes são exclusivas para funcionários e empregados e esquecem a importância que é oferecer um canal de comunicação mais pessoal com o cliente final. Isto permite não só captar a atenção e interesse das pessoas como também permitir oferecer um serviço mais personalizado e agradável para cada tipo de cliente [1].

Com o produto a desenvolver no âmbito deste projecto, pretende-se dar resposta aos problemas explicitados e ir mais além, permitindo um sistema integrado, envolvendo novas ferramentas de *Software* e a utilização de novo *Hardware*, que proprietários, empregados e clientes podem usar em seu benefício.

## 1.2. EQUIPA

Dado tratar-se de um projecto que nasceu numa *start-up*, a equipa responsável pelo desenvolvimento do projecto proposto é relativamente pequena. As tarefas de especificação da aplicação e implementação da solução proposta ficaram ao encargo do estagiário Diogo Dias e de um outro elemento sénior, mais precisamente o Eng.º Rui

Teixeira. Por outro lado, o Eng.º Vasco Pereira assumiu as funções de coordenador técnico do projecto (como consultor da Universidade de Coimbra) e orientador de estágio, simultaneamente. Não posso deixar de referir ainda a preciosa ajuda facultada pelo Miguel Faria (CEO da empresa Softkom2) e dos seus colaboradores, que nos auxiliaram especialmente na fase de especificação do produto, através da partilha de experiências reais do dia-a-dia e respondendo a dúvidas que fossem aparecendo relacionadas com o nosso problema. Por fim convém só mencionar que o elemento Miguel Faria assumiu igualmente o papel de cliente no contexto deste projecto.

### 1.3. METODOLOGIA DE DESENVOLVIMENTO

Neste capítulo irei explicar de forma sucinta a forma escolhida para a realização do projecto e as razões que levaram a esta abordagem. Para além disso, será também dado a conhecer exactamente o tipo de tarefas e responsabilidades que foram atribuídas ao estagiário.

O projecto SIGER tem as características de ser um projecto inovador, não requerendo a necessidade de integrar com qualquer tipo de aplicação desenvolvida previamente. Contudo, isto também obriga a um maior cuidado na análise e planeamento do produto, pois existem sempre muitas questões iniciais em aberto. Por essa razão e devido à dimensão da equipa, ficou desde logo claro que seguir um modelo de desenvolvimento rígido, não iria trazer grandes vantagens para o sucesso do projecto. Por essa mesma razão, e dado que nenhum dos elementos da equipa possuía conhecimento do negócio da restauração, optou-se desde o início por investir consideravelmente na fase de levantamentos de requisitos e especificação da aplicação a desenvolver. Este tempo investido iria permitir a familiarização com o ‘*modus operandi*’ nesta área de negócio e ao mesmo tempo garantir que as expectativas do cliente estariam a ser interpretadas correctamente. Para que isso fosse possível, foi necessário a estreita colaboração com o actor ‘Cliente’, através de reuniões semanais (ou quinzenais) para validar o trabalho realizado.

Mais tarde, e depois de concluído o levantamento de requisitos, seguiu-se para a fase de desenho da arquitectura e da especificação dos diagramas de classes e de entidade e relacionamento. Esta fase contou com vários ciclos de revisão por forma a refinar detalhes técnicos da arquitectura ou nos diagramas produzidos.

Finda a fase anterior, entramos na parte de implementação da solução proposta. Antes disso foi ainda necessário voltar à lista de requisitos produzida por forma a definir claramente as funcionalidades passíveis de serem realizadas dentro do tempo disponível para a conclusão do projecto. Desta análise resultou um conjunto de requisitos que se tornaram o *backlog* do projecto. Convém referir que para a fase de desenvolvimento optou-se por seguir uma metodologia mais ágil. Mais concretamente foi escolhida a metodologia *Scrum* e as razões prendem-se com o facto deste tipo de metodologia permitir uma gestão flexível do projecto, baseado em iterações de muito curta duração e centradas na equipa, de forma a ter um controlo próximo sobre trabalho desenvolvido e ao mesmo tempo conseguir mostrar um produto preliminar ao cliente que vai sendo aprimorado ao longo das várias entregas. Para além disso, este tipo de metodologia oferece alguma flexibilidade que permite acautelar pequenas alterações que possam vir a surgir. A fase de implementação inclui igualmente a realização de testes de unidade e de



integração entre componentes desenvolvidos, de forma a garantir um padrão de qualidade a cada *sprint* de desenvolvimento.

No entanto é preciso lembrar que a metodologia *Scrum* aplicada no contexto deste projecto não seguiu o formato tradicional mas foi alvo de algumas adaptações para se ajustar à dimensão e características da equipa. A implementação iniciou com um *backlog* detalhado e com *sprints* projectados para 15 dias uteis, onde por sua vez cada *sprint* teve direito a uma reunião prévia de modo a estimar o esforço por tarefa e que tarefas fariam parte do *sprint*. Para além dos *sprints* comuns que incluíam a resolução das tarefas do *backlog*, foram criados alguns *sprints* extraordinários de forma a corrigir problemas detectados durante a integração das funcionalidades produzidos nos *sprints* comuns. Outro aspecto que não foi totalmente seguido à regra foram as tradicionais ‘*stand-up meetings*’. Neste aspecto a troca de ideias e discussão de problemas foi realizada de forma mais natural e informal, visto que a equipa técnica era constituída apenas por 3 elementos.

### Lista de tarefas e responsabilidades

Análise e Especificação dos Requisitos	Todos os elementos foram convidados a participar activamente no levantamento e análise dos requisitos, cabendo depois a tarefa de escrita da documentação, ao estagiário e ao Eng.º Rui Teixeira.
Análise e Especificação da Arquitectura	Esta tarefa seguiu o mesmo modelo da tarefa anterior. Todos os elementos foram convidados a participar activamente, cabendo depois a tarefa de escrita da documentação, ao estagiário e ao Eng.º Rui Teixeira.
Análise e escolha das ferramentas de desenvolvimento	Esta tarefa foi desempenhada pelos elementos da equipa de desenvolvimento que consiste no estagiário e no Eng.º Rui Teixeira.
Teste de exploração de algumas tecnologias envolvidas no projecto	Esta tarefa foi realizada principalmente pelo estagiário e permitiu perceber as possibilidades e limitações dos equipamentos que irão ser usados numa fase mais adiantada do projecto. Mais concretamente, foram realizados testes com os dispositivos <i>Drink-Dispensers</i> com vista a perceber o seu funcionamento e a forma usada para controlá-los usando a biblioteca ‘ <i>Skyman.dll</i> ’.
Gestão do projecto	Esta tarefa ficou ao encargo do Eng.º Vasco Pereira. O estagiário participou activamente nas reuniões semanais da equipa e na preparação dos <i>sprints</i> da fase de implementação.
Implementação	Nesta fase o conjunto de funcionalidades que compuseram o <i>backlog</i> do projecto foram divididas de igual forma e distribuídas pelo estagiário e pelo outro elemento sénior da equipa. De entre as várias tarefas desenvolvidas pelo estagiário podemos salientar as seguintes onde ele participou activamente:

	<ul style="list-style-type: none"> <li>• Funcionalidades de gestão de utilizadores</li> <li>• Funcionalidades de gestão de grupos de utilizadores e permissões de acesso.</li> <li>• Comunicação com impressoras e impressão de documentos usando ESC/P (Epson Standard Code for Printers)</li> <li>• Estudo e desenho da UI</li> <li>• Controlo e gestão de serviços efectuados nos Drink-Dispensers.</li> <li>• Funcionalidades de Registo e Pagamento</li> </ul>
Testes	<p>A realização dos testes unitários ficou ao encargo de cada programador, sendo cada um responsável por garantir a qualidade do seu código. Já na fase dos testes funcionais optou-se por trocar os papéis de modo a que cada programador testasse os módulos desenvolvidos pelo outro parceiro.</p> <p>Nesta tarefa incluem-se também ainda os testes de aceitação e <i>pré-realase</i> realizados junto do cliente e por agentes da empresa no terreno.</p>

#### 1.4. PLANEAMENTO

O planeamento inicial foi realizado para o período de 03 de Setembro até 28 de Junho. De uma forma muito genérica, previa-se que no primeiro semestre o estagiário tivesse um primeiro contacto com a empresa e o projecto, e ao mesmo tempo pudesse participar na especificação da documentação de apoio ao desenvolvimento do projecto. No segundo semestre estava então planeado o arranque do desenvolvimento do projecto, de acordo com o trabalho produzido no semestre anterior.

De acordo com a proposta de estágio inicial, pretendia-se desenvolver um conjunto de módulos autónomos que integrados entre si, seriam a base do projecto SIGER. A estratégia inicial era que estes módulos iriam trabalhar sobre um sistema de facturação já existente de forma a facilitar a sua introdução no mercado actual. Contudo, por motivos internos à empresa e aos seus parceiros, não foi possível seguir essa abordagem pelo que o plano de trabalhos teve de ser alterado. Por essa razão e dado a importância do módulo de POS/Facturação para o resto do projecto, fomos obrigados a incluir este requisito na especificação do projecto. Sendo assim, o âmbito deste estágio teve por base a especificação e desenvolvimento do módulo POS e do módulo de controlo dos *Drink-Dispensers*. Isto significa que a tarefa 'Integração com *PingWin*', constante no plano inicial do estágio, foi removida do plano de trabalhos e substituída com as tarefas de 'Análise e Desenvolvimento do Módulo POS'.

Outra tarefa que constava da proposta inicial do estágio, e que não foi passível de ser realizada durante este estágio, foi a 'Definição de uma interface conjuntamente com equipas externas'. Esta tarefa visava desenvolver em sintonia com o LAS (*Laboratory of Automatics and Systems*), uma API de comunicação com os dispositivos RFID/NFC que viriam a ser usados futuramente nalguns módulos da aplicação. A razão para tal, prende-

se com o facto de os trabalhos desenvolvidos nessa entidade externa, terem sofrido atrasos e como tal, condicionou a realização desta tarefa. Isto não quer dizer que a tarefa tenha sido esquecida ou removida do plano de trabalhos, apenas significa que teve de ser adiada e que como tal não pôde ser desenvolvida no intervalo de tempo em que decorreu o estágio académico.

Convém realçar que o estagiário iniciou o seu trabalho na sede da empresa em Julho, no entanto o planeamento apresentado representa apenas as tarefas que foram executadas no contexto do estágio académico. Contrariamente ao que é usual neste tipo de estágios, o estagiário esteve envolvido a tempo inteiro na realização suas funções em prol da empresa, logo a partir do primeiro semestre.

Mais à frente são apresentados o mapa de Gantt projectado no início do projecto e ainda o mapa de Gantt real. Como é fácil de verificar existe uma certa discrepância entre ambos os planos. As razões que obrigaram a este atraso prendem-se com a necessidade de incluir alguns ajustes e pequenas funcionalidades novas de modo a fornecer resposta a situações especiais entretanto apresentadas pelo cliente (tais como recuperação de facturas manuais, revisão da UI geral da aplicação, controlo de acesso/permisões, gestão de mesas, etc.). Outro factor não menos importante, teve a ver com a necessidade de alocar mais tempo do que o esperado para corrigir bugs detectados durante as fases de testes de integração realizadas.

As alterações introduzidas este ano à legislação no que diz respeito aos sistemas de facturação obrigaram a um cuidado redobrado no processo de implementação do módulo POS. Para além disso, ao longo do processo de desenvolvimento fomos igualmente surpreendidos com correcções sucessivas à lei que acabaram por obrigar a alterações não previstas no plano inicial.

Devido a todos estes atrasos e de modo a conseguir concluir as tarefas inicialmente planeadas para este estágio, optou-se por prolongar os trabalhos até à época especial de Setembro. Essencialmente, o tempo adicional foi aplicado na implementação das funcionalidades base especificadas para o módulo de controlo de *Drink-Dispensers* e ao mesmo tempo integrar essas funcionalidades com o módulo de facturação produzido antes.

Por fim, outra tarefa que não seguiu totalmente o plano traçado inicialmente foi a escrita do manual de utilizador da aplicação. Este documento, contrariamente ao que tinha sido planeado, foi sendo escrito à medida que as várias funcionalidades iam sendo implementadas ou ainda quando surgia a necessidade de proceder a alterações na especificação original.

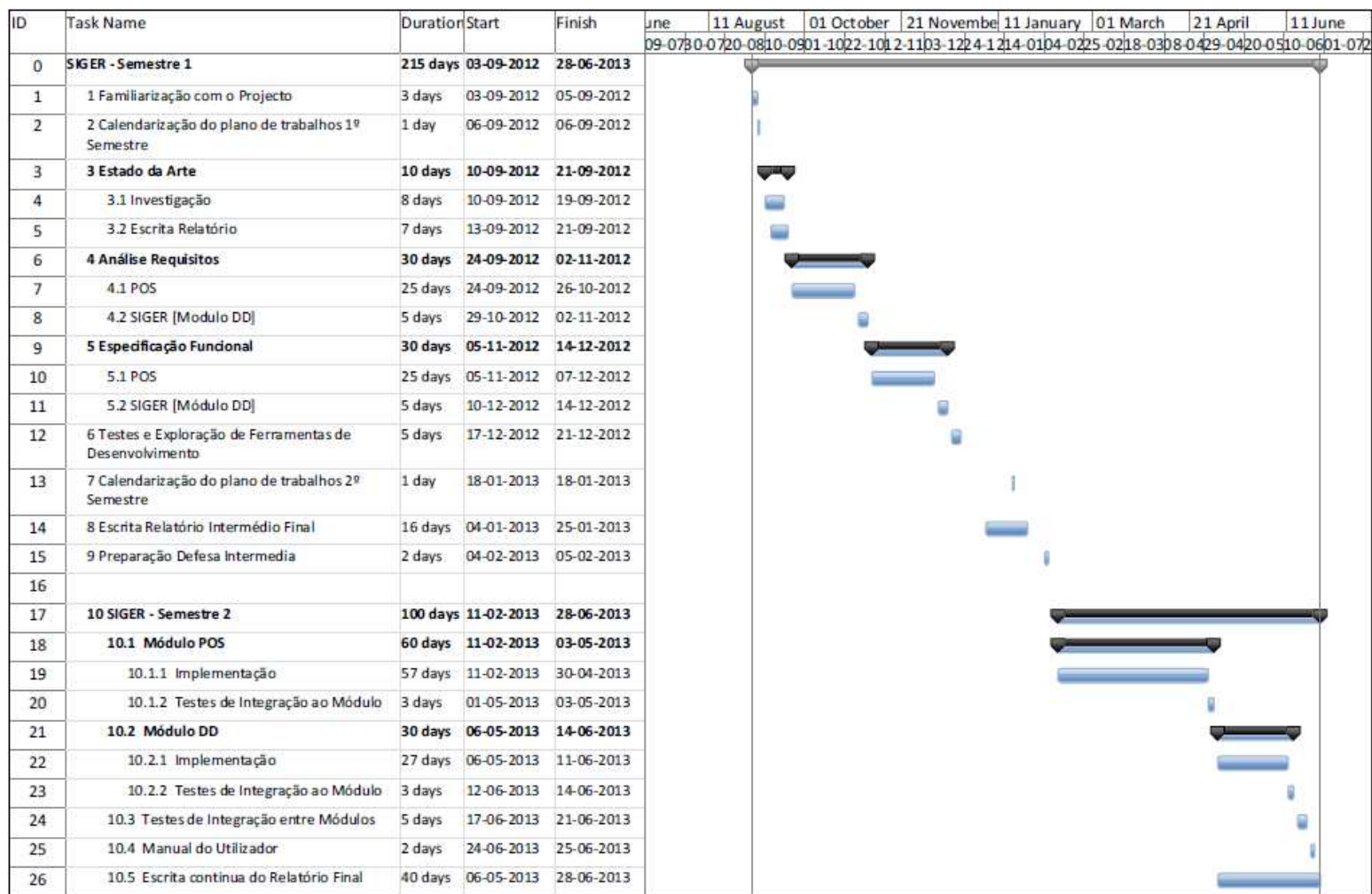


Figura 1 - Calendarização do plano de trabalhos [versão planeada]

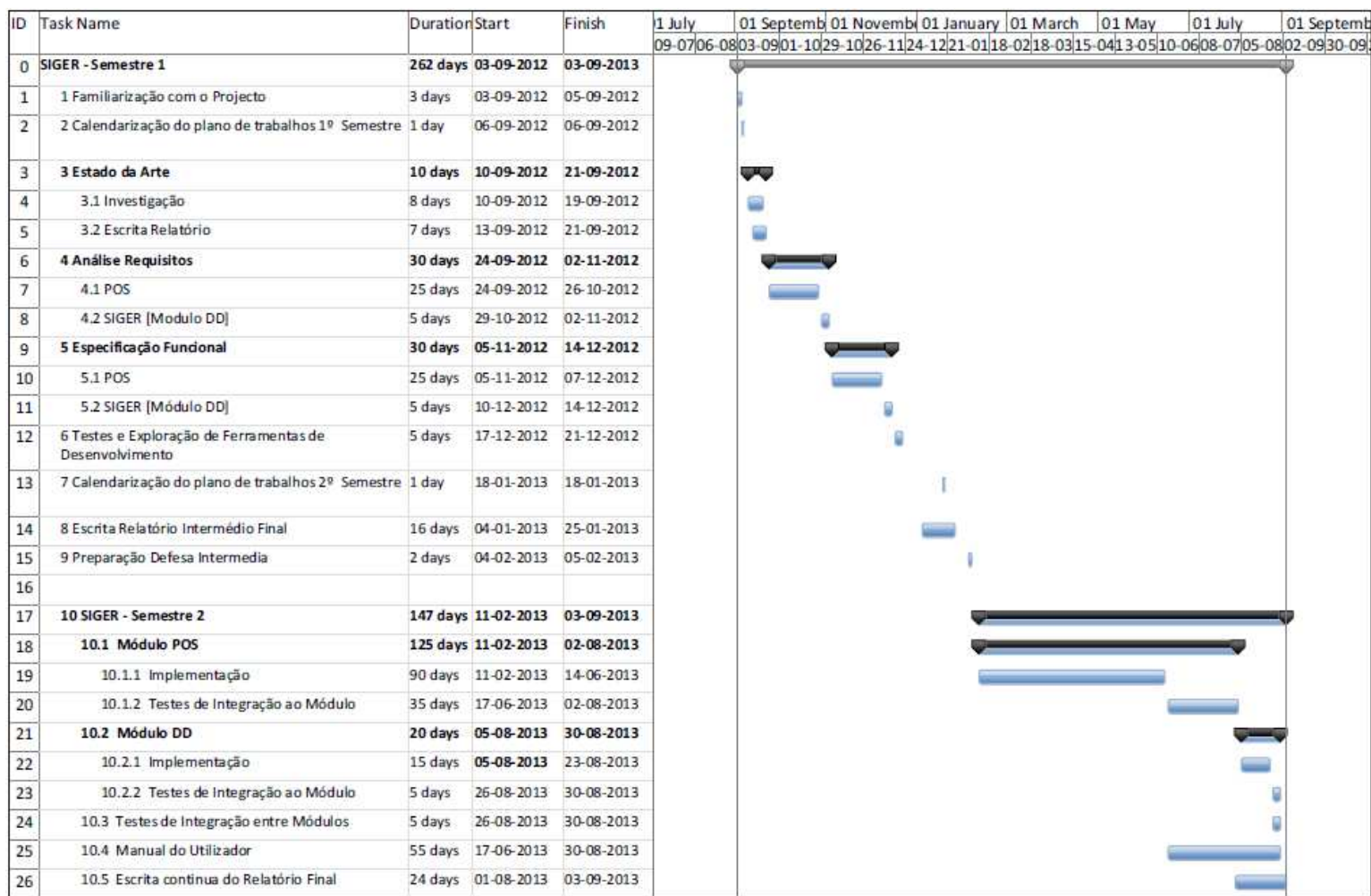


Figura 2 - Calendarização do plano de trabalhos [versão real]

## 2. ESTADO DA ARTE

Neste capítulo será apresentado um estudo detalhado dos principais produtos concorrentes existentes actualmente no mercado, e será igualmente feita uma análise às principais tecnologias que irão integrar o produto final.

O estado da arte foi realizado em duas fases distintas. Numa primeira parte (primeiro semestre) foi dado ênfase às funcionalidades oferecidas pelos produtos concorrentes. Esta fase visou sobretudo perceber quais as vertentes em que o produto era capaz de se diferenciar dos restantes. Já numa segunda fase (segundo semestre) a análise debruçou-se exclusivamente sobre o aspecto visual e usabilidade dos produtos concorrentes. O objectivo desta segunda fase foi perceber qual o caminho seguido pelos nossos concorrentes na área da usabilidade e como poderíamos melhorar essa vertente no nosso produto.

### 2.1. PRODUTOS CONCORRENTES

A análise do estado da arte realizada neste capítulo, tem por base produtos concorrentes com funcionalidades de Ponto de Vendas (POS), orientados para a área da restauração/retalho. Após um breve resumo de cada um dos produtos concorrentes identificados, será ainda dado espaço a uma análise comparativa às suas principais funcionalidades.

#### 2.1.1. WINREST FRONTOFFICE PRO

Este produto nasceu no ano de 1995 dentro do grupo PIE [2] e foi rapidamente ganhando protagonismo no mercado de retalho nacional e internacional. O **WinRest** [3] é neste momento líder de mercado em Portugal estando também presente em vários outros países da Europa, África e América. A aplicação é constituída por vários módulos, que podem ir sendo acrescentados de acordo com as necessidades do cliente. Enumerando apenas alguns desses módulos, temos por exemplo o módulo de fidelização de clientes, o módulo de Vídeo Vigilância, módulo de *WebReports* ou ainda o módulo de integração com máquinas de café. Além dos terminais de posto de venda, suporta também comandos via rádio ou a integração com PDAs.

Mais tarde, e após se aperceber da grande procura em vigor na área de retalho de soluções para gestão de negócio, o grupo PIE decidiu enveredar por uma abordagem diferente e daí nasceu o **PingWin** [4]. Este produto ao contrário do **WinRest** não se restringe à restauração mas tenta integrar no mesmo produto vários ramos do sector de retalho. A solução encontrada para este problema passou por manter e evoluir o core do **WinRest** e construir-lhe uma camada gráfica por cima que pudesse ser personalizada de acordo com cada tipo de negócio. A PT Negócios é actualmente um aliado de peso para a massificação destes dois produtos no território nacional.

### 2.1.2. ZSREST FRONTOFFICE

O **ZsRest** [5] é mais um produto nacional bastante completo oferecido pela empresa ZoneSoft. A abordagem desta empresa passou igualmente por disponibilizar diferentes versões para cada uma das principais áreas da restauração, ou seja, são actualmente disponibilizadas as aplicações ZsRest, ZsCafé e ZsPos. Entre a vasta lista de funcionalidades oferecidas pela aplicação, gostaria de realçar as seguintes:

- Possibilidade de suspender/recuperar uma conta
- Aceita vários tipos de pagamento (dinheiro, MB, conta-corrente)
- Multi-utilizador com gestão de entrada/saída
- Edição das permissões por utilizador
- Acesso a relatórios estatísticos
- Interoperabilidade com vários dispositivos externos tais como balanças, leitores de códigos de barras, impressora de talão e A4, etc.

No quadro comparativo irei ter em conta apenas as funcionalidades da solução **ZsRest** visto ser a solução mais completa do grupo [6]. Apesar de a aplicação ser bastante completa, o seu impacto visual e usabilidade podem ser algo confusos e introduz uma curva de aprendizagem elevada ao consumidor menos experiente. Em certas janelas das aplicações os elementos gráficos de interacção com o utilizador não são os mais apropriados para utilizar em ambientes *touch*.

### 2.1.3. SAGE GESREST II

A solução **GesRestII** [7] da Sage é mais um produto português direccionado para terminais POS na área da restauração. A aplicação **GesRestII** consiste numa aplicação adaptável com módulos específicos para restaurantes, cafetarias, pastelarias, bares e discotecas. Uma grande vantagem deste produto reside na vasta gama de plataformas que suporta desde sistemas Windows até plataformas móveis com Android. Para além disso nota-se que existiu um cuidado maior no desenho e usabilidade gráfica desta aplicação quando comparado com os outros concorrentes.

### 2.1.4. GESTWIN POS

A empresa SoftPack entrou no mercado nacional em 1992 com o seu produto **GestWin POS** [8] dirigido ao pequeno retalho, para gestão comercial em POS e baseado nas plataformas Windows. Neste momento oferece diferentes versões do seu produto para cada uma das áreas de retalho possíveis. As várias versões partilham um conjunto de funcionalidades base entre si, possuindo depois especificidades extras para cada tipo negócio. De entre as várias funcionalidades oferecidas por este produto seria interessante realçar as seguintes:

- Possibilidade de apurar encomendas necessárias ao fornecedor
- Exportação de dados
- Transferência de mesas entre funcionários
- Reserva de mesas

- Gestão de *Happy-Hours*
- Relatórios estatísticos de indicadores relevantes ao negócio (por exemplo: Cálculo da Rentabilidade, historial de preços, controlo de validades, etc.)

No quadro comparativo irei ter em conta apenas as funcionalidades da solução “GestWin POS Universal” visto ser a solução que inclui as funcionalidades base de todas as outras versões.

### 2.1.5. HARBORTOUCH

A **HarborTouch** [9] é uma empresa norte americana especializada na prestação de serviços e venda de soluções para a área da restauração. Segundo a própria, estima-se que os seus produtos estejam instalados em cerca de 110.000 postos de trabalho e facturam anualmente um total de 9 mil milhões de dólares. O seu modelo de negócio aposta exclusivamente na venda de sistemas completos, ou seja, *Hardware & Software*. O sistema HarborTouch apresenta todo o tipo de funcionalidades essenciais nas áreas do retalho e restauração, aliados a uma interface simples e elegante. Para além disso é possível actualmente instalar a aplicação nos *tablets* iPad e integrar estes terminais no processo de trabalho da restante estrutura de pontos de venda do estabelecimento.

### 2.1.6. QUADRO COMPARATIVO

Na Tabela 1 é apresentado um quadro comparativo das diferentes funcionalidades dos produtos concorrentes e da solução que o estagiário se propõe a desenvolver.

	WinRest FO	ZsRest FO	Sage GesRest II	GestWin POS Universal	HarborTouch	SIGER
Ambiente <i>Touch</i>	✓	✓	✓	✓	✓	✓
Plataforma	Windows / Linux	Windows	Windows / Linux / Android	Windows	Linux/ iOS	Windows / Linux
Suporte Multilingue	✓	✗ (Português)	✗ (Português)	✗ (Português)	✗ (Inglês)	✓
Integração com <i>hardware</i> externo (impressoras, caixas automáticas, balanças, comandos rádio X64, display externos, balanças)	✓	✓	✓	✓	✓	✓
BackOffice	✓	✓	✓	✓	✓	✓
Acesso e controlo via WEB	✗	✓	✓	✗	✓	✓



Módulo de Gestão de Clientes	✓	✓	✓	✓	✓	✓
Módulo de Gestão de Promoções e Happy-Hours	✓	✓	✓	✓	✓	✓
Módulo de Gestão de Stock, Inventários e Compras	✓ (módulo extra)	✓	✓	✓	✓	✓
Rastreamento de Stocks através de RFID	✗	✗	✗	✗	✗	✓
Módulo de gestão de Funcionários e permissões	✓	✓	✓	✓	✓	✓
Módulo de Gestão de Artigos	✓	✓	✓	✓	✓	✓
Módulo de Relatórios com indicadores sobre o negócio	✓ (módulo extra)	✓	✓	✓	✓	✓
Sistema de Notificações	✗	✗	✗	✗	✗	✓
Integração Multi-Posto	✓	✓	✓	✓	✓	✓
Funcionamento em modo <i>offline</i> (em caso de falhas na rede)	✓	✓	✓	✓	✓	✓
Integração com leitores código barras	✓	✓	✓	✓	✓	✓
Integração com leitores RFID / NFC	✗	✗	✗	✗	✗	✓
Troca de Mensagens entre Terminais	✗	✓	✗	✗	✓	✓
Interface de Integração com Dispensadores de Bebidas	✗	✓	✗	✗	✗	✓

Tabela 1 - Quadro comparativo Produtos Concorrentes

A partir da tabela 1 temos acesso às principais características que diferenciam o nosso produto dos da concorrência. Uma característica que salta imediatamente à vista é a falta de suporte para várias línguas na maioria das aplicações. Isto é uma falha que pode impedir a expansão do produto para mercados internacionais.

Hoje em dia, com os avanços na computação distribuída, sobretudo móvel, é dada cada vez mais importância à necessidade de poder consultar ou gerir os negócios a partir

de qualquer lugar no mundo. Como tal, o acesso Web será sem dúvida um trunfo importante do nosso produto, e que continua a faltar em alguns concorrentes directos.

Um outro ponto que distingue o nosso produto da concorrência, é a aplicação da tecnologia RFID para a realização de inventários e gestão de *stocks* sobre produtos equipados com etiquetas RFID, ou ainda a aplicação da tecnologia NFC para lançamento campanhas promocionais interactivas junto dos clientes. Este novo paradigma da interacção com o cliente torna-se uma mais-valia importante, se ao mesmo tempo conseguirmos com base em históricos do cliente, oferecer um serviço mais direccionado aos hábitos de consumo e preferências do mesmo. Para além disso, esperamos que com a introdução de serviços de pagamento via NFC num futuro próximo (por exemplo 'TMN Wallet' [10]), consigamos manter vantagem tecnológica e de inovação em relação aos nossos concorrentes directos. Por fim, com a introdução das tecnologias RFID/NFC pretende-se igualmente cortar nos custos de fabrico de cartões e/ou talões recorrendo única e simplesmente aos dispositivos móveis dos clientes.

O nosso sistema irá possuir de raiz um mecanismo de gestão de notificações não intrusivo para notificar os utilizadores sobre alterações relevantes na aplicação ou dispositivos interligados. Esta funcionalidade irá permitir manter o utilizador informado sobre acontecimentos importantes relacionados com o sistema, sem para isso interromper o processo de trabalho dos utilizadores. Para complementar esta funcionalidade, será aplicado um mecanismo de troca de mensagens entre terminais para uma comunicação rápida entre utilizadores. Ambas as funcionalidades apresentadas são novidades quando comparadas com os produtos alternativos.

Apesar de todos os produtos concorrentes apresentados possuírem de forma mais ou menos completa, funcionalidades de apoio ao negócio tais como relatórios estáticos de vendas, *stocks* de produtos, indicadores de desempenho dos funcionários, e etc., a verdade é que muitas das vezes essas funcionalidades são completamente descartadas pelos gestores, principalmente devido à total falta de relevância para a realidade do dia-a-dia do estabelecimento. Por essa simples razão, pretende-se que o produto SIGER disponibilize de forma simples e directa (sobre a forma de alertas), os indicadores mais relevantes à gestão do negócio evitando ao máximo 'bombardear' o utilizador com dados estatísticos difíceis de decifrar. Só numa segunda fase e se o utilizador assim o desejar, será possível consultar mapas estatísticos completos sobre indicadores tais como por exemplo o cálculo do lucro do dia/semana/mês, historial dos produtos consumidos, historial das vendas por empregado, produtos há mais tempo em *stock*, produtos com validade a expirar e etc. O acesso a este tipo de indicadores será realizado através do módulo de gestão de negócio que está pensado para o projecto SIGER mas que não fará parte do âmbito do estágio.

Por fim, o SIGER irá, comparativamente com os concorrentes directos, integrar um sistema de controlo de bebidas capaz de produzir relatórios informativos sobre os movimentos realizados no estabelecimento e permitir rentabilizar ao máximo os proveitos do cliente. Apesar de existir um produto concorrente (ZsRest) que afirma estar preparado para integrar este tipo de equipamentos, a verdade é que no terreno ainda não se encontram soluções destas implementadas.

Por todas estas razões, acreditamos que as inovações apresentadas pelo SIGER serão um factor de diferenciação e valorização da marca no território nacional e internacional.

Dos vários produtos analisados na fase de estudo da usabilidade dos produtos concorrentes, tiveram maior relevo o Pssst! Primavera [11] (nacional) e o Square Register [12] (internacional). Dado que ambos estes produtos surgiram no mercado já depois do estudo do estado da arte realizado no primeiro semestre, e visto que nenhum deles traz algum tipo de inovação ao nível das funcionalidades oferecidas optou-se por não incluir estes produtos no quadro comparativo. No entanto ao nível de usabilidade foi possível retirar algumas ideias interessantes, presentes em ambos os produtos:

- Componentes interactivos com proporções ideais para ambientes *touch*
- Personalização de cores e tamanhos de letras
- Conjunto de elementos interactivos por ecrã limitado
- Flat Design [13]

## 2.2. TECNOLOGIAS ENVOLVIDAS

Neste capítulo serão apresentados brevemente as principais tecnologias aplicadas no âmbito do projecto proposto. A ideia é mostrar de que forma a sua aplicação pode valorizar o negócio dos clientes e dar a conhecer as suas principais características.

### 2.2.1. SISTEMAS DE CONTROLO DE BEBIDAS

A ideia por trás desta solução é conseguir rentabilizar ao máximo as bebidas em *stock* no estabelecimento, através dos seguintes princípios:

- Medidas consistentes das doses servidas
- Reduzir sobredosagens e desperdícios
- Apurar responsabilidades visto que todos os serviços realizados são registados no sistema de controlo e associados a um empregado
- Evitar ofertas não registadas no sistema
- Aumentar os proveitos através do controlo de custos
- Ao servir doses consistentes é possível identificar mais claramente a quantidade de álcool ingerido pelos clientes e bloquear o serviço a clientes que tenham excedido os limites definidos

A nível nacional não existem concorrentes que incorporem este tipo de tecnologias nos seus produtos. Daí percebemos que esta pode ser um trunfo importante para a entrada do nosso produto neste mercado. Já a nível internacional existem alguns fabricantes deste tipo de dispositivos que fornecem soluções *stand-alone* com *software* para controlar os seus dispositivos, mas que não permitem uma integração fácil com os sistemas de facturação (POS) actuais.

### 2.2.2. RFID E NFC

Tanto a tecnologia **RFID** (*Radio Frequency IDentification*) como a tecnologia **NFC** (*Near Field Communication*) são fundamentalmente tecnologias de troca de informação sem fios baseadas em frequências rádio. A tecnologia NFC é uma evolução natural da RFID onde a grande diferença entre ambas reside no facto de ter existido um maior

cuidado na elaboração de *standards* na concepção do NFC. Isto permite uma maior interoperabilidade entre dispositivos de fabricantes diferentes. De forma genérica, dispositivos NFC e RFID típicos podem coexistir num mesmo cenário visto que têm aplicações diferentes.

A tecnologia RFID consiste numa etiqueta electrónica munida de uma antena e de um chip de memória que quando activada é capaz de enviar a informação presente na sua memória para o dispositivo leitor [14]. Usualmente as etiquetas são activadas pelo leitor (modo passivo) porém também existem etiquetas mais complexas que incorporam bateria e desse modo não precisam do leitor para veicularem a sua informação (modo activo). As etiquetas activas são sobretudo usadas em cenários que exigem a leitura de dados de sensores ou algum tipo de processamento extra por parte da etiqueta. Dada a maior complexidade técnica das etiquetas activas, estas são substancialmente mais caras e maiores do que as etiquetas passivas. A utilização de um ou outro tipo de dispositivos prende-se sobretudo com o cenário ao qual devem ser aplicadas.

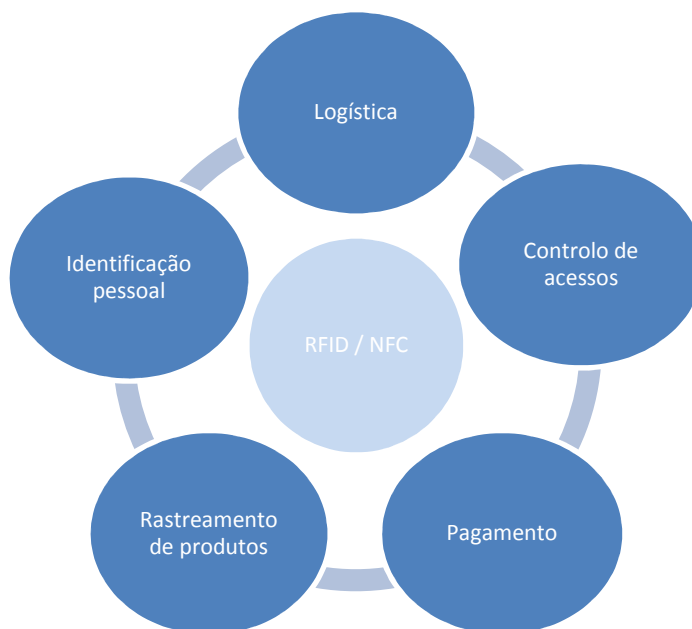


Figura 3 - Possibilidades de aplicação das tecnologias RFID/NFC

A tecnologia NFC disponibiliza um raio de comunicação bastante reduzido (até 10 cm) quando comparado com a RFID (vários metros dependendo do tipo de etiquetas e leitores). A razão principal prende-se com a segurança. Para além disso, outra diferença entre ambas reside no facto que a RFID só permite comunicação no sentido etiqueta RFID -> Leitor RFID, enquanto na tecnologia NFC pode existir comunicação bidireccional (*Peer 2 Peer*) [15].

Focando-nos na área de actuação do produto SIGER, actualmente as principais áreas de aplicação das tecnologias RFID/NFC na restauração são fundamentalmente:

- Sistemas de Controlo de acessos – Usado para identificação e/ou autenticação de empregados, clientes ou fornecedores, nos sistemas de informação disponibilizados no estabelecimento.

- Sistemas de Pagamento – Visa oferecer uma nova forma rápida e simples de pagar por artigos/serviços consumidos recorrendo simplesmente ao telemóvel do cliente [10].
- Sistemas de Rastreamento de pessoas/objectos – Usado no controlo de *Stocks* ou encomendas, e na fase de venda dos produtos.
- Sistema de Quiosque e gestão de informação - Este tipo de funcionalidade consiste usualmente em fornecer um canal de acesso personalizado a informações sobre produtos e serviços, através da comunicação NFC com os dispositivos móveis dos clientes.

Apesar de já existirem actualmente alguns equipamentos (*smartphones* e *tablets*) à venda no mercado actual equipados com NFC, o facto de estes serem em número reduzido está a dificultar a entrada desta tecnologia neste sector. Contudo existe já actualmente uma grande investimento nesta área por se acreditar que esta pode revolucionar a forma de interagir com os clientes nas mais variadas vertentes de negócio. A nível nacional temos o exemplo da PT que está actualmente em fase de testes da versão piloto da sua aplicação “*TMN Wallet*” [10] para a realização de pagamentos simples via NFC. De acordo com a previsão de vários especialistas, prevê-se que dentro de 1 ou 2 anos, a NFC venha a ter um protagonismo predominante no dia-a-dia das pessoas [16].

### 2.2.3. CASO DE ESTUDO

O ‘*Tavern Restaurant Group*’ [17] opera uma rede de restaurantes e *pubs* tipicamente ingleses espalhados pelo centro oeste e sul do Estados Unidos. Em Novembro do 2010 o grupo abriu um novo estabelecimento em Tampa na Florida e a partir de uma colaboração com a empresa ‘*DraftServ Technologies*’ foi instalado no estabelecimento um sistema único de monitorização e gestão das bebidas incluindo uma zona inovadora de *self-service* para os seus clientes. O sistema de monitorização consiste num conjunto de válvulas cuidadosamente instaladas entre as fontes (barril de cerveja) e as torneiras de saída que captam as quantidades exactas de líquidos servidas e enviam essas leituras para o servidor central. O servidor está acessível através da WEB e como tal o dono do estabelecimento consegue avaliar em qualquer momento e em qualquer lado os movimentos realizados localmente. Para além das várias torneiras regulares usadas pelos funcionários para servir os clientes, existe igualmente uma zona com 20 torneiras de diferentes tipos de cerveja exclusivamente dedicadas ao uso pessoal por parte dos clientes. O controlo é feito através um cartão RFID (tipo passivo) que é entregue à entrada. Antes disso o cliente precisa de se identificar por forma a perceber se este é menor de idade ou não. Caso seja maior de idade é lhe atribuída um determinado *plafond* de volume liquido permitido, que o cliente pode consumir da forma que lhe bem apetece. Para efectuar um serviço de uma determinada cerveja basta aproximar o seu cartão junto à torneira desejada e servir-se abrindo a torneira indicada (temporizador de 20 segundos entre identificação e serviço). O sistema irá automaticamente contabilizar o volume de cerveja servido desde o momento que abriu a torneira até ao momento de fecho da torneira. O *plafond* permite controlar o estado de embriaguez dos clientes e se o funcionário perceber que o cliente já não está em condições de consumir mais, ser-lhe-á bloqueado o acesso às torneiras.

O cartão RFID serve exclusivamente de identificador estando toda a lógica a ser controlada pelo servidor e pelo sistema de monitorização instalado no estabelecimento. É igualmente possível aos utilizadores informarem-se acerca das características de cada cerveja e do seu histórico de consumos utilizando terminais (*iPad*) fixados na parede ou averiguar estatísticas das bebidas com mais saída que são apresentadas em ecrãs de grande dimensão no estabelecimento.

Em termos de retorno de investimento, o grupo rapidamente concluiu que em média o número de bebidas por barril neste *pub* é superior aos seus outros estabelecimentos, em cerca de 16-20%. Em média, neste tipo de negócios e sem utilizar nenhum tipo de sistema de monitorização, é comum conseguir-se aproveitar cerca de 75% da capacidade do barril. Os outros 25% são desperdiçados ao servir, oferecidos ou doseados em excesso. O sistema inovador de *self-service* mostrou ainda um aumento em cerca de 14% das vendas comparativamente com o serviço tradicional. Ao ter acesso detalhado aos números de vendas de cada funcionário foi possível identificar discrepâncias entre o que foi servido e o que foi registado no sistema e desta forma obter um maior retorno por bebida servida.

Através das conclusões apresentadas neste caso de estudo, rapidamente nos apercebemos das mais-valias oferecidas por uma solução deste tipo. Não só é possível manter um controlo mais apertado de todas as transacções realizadas como ao mesmo tempo rentabilizar mais eficazmente os produtos oferecidos dentro do estabelecimento. Com a introdução deste tipo de soluções é possível diferenciar-se da concorrência e ao mesmo tempo apelar à curiosidade da clientela.

#### 2.2.4. CLOUD VS LOCAL

Uma questão pertinente para colocar é, dada a grande popularidade actual dos serviços *Cloud*, porque não apostar numa abordagem SaaS (Software as a Service) em detrimento de uma instalação no local?

Antes de poder responder a essa questão convém lembrar primeiro as principais vantagens e desvantagens oferecidas pelo modelo SaaS (Software as a Service).

Vantagens [18]:

- Eficiência de custos – Dado que este tipo de modelo se baseia na prestação de serviços via web e como tal grande parte do poder de processamento é suportado pela *Cloud*, isto permite eliminar parte do investimento inicial necessário em *Hardware* especializado.
- Disponível em qualquer lugar – Por se tratar de um serviço *Web*, o acesso às funcionalidades oferecidas passa a estar disponível em qualquer parte do mundo desde que o utilizador tenha um acesso à Internet.
- Actualizações instantâneas – Sempre que o *Software* disponibilizado for actualizado, estas alterações são rapidamente propagadas para os vários clientes sem necessidades de intervenção de técnicos especializados.
- Escalabilidade – Usualmente, sempre que exista a necessidade de expandir a capacidade de processamento ou armazenamento na *Cloud*, este problema pode ser facilmente resolvido através da contratação de melhores condições junto do prestador de serviço. O valor de investimento é normalmente menor

do que se fosse feito um investimento directo em novo *Hardware* e mão-de-obra de técnicos especializados.

Desvantagens:

- Segurança e privacidade – Ao enveredar pelo modelo em *Cloud*, o utilizador está basicamente a disponibilizar toda a sua informação pessoal e a confiar totalmente no prestador do serviço web. O utilizador não só está a fornecer toda a sua informação confidencial como também é obrigado a aceitar que essa informação não está totalmente protegida de acesso por parte de terceiros.
- Quebras no acesso – Problemas no acesso à Internet ou mesmo nas infra-estruturas do prestador de serviço na *Cloud*, pode obrigar a paragens no negócio por impossibilidade da utilização correcta da aplicação.
- Limitações técnicas – Devido ao facto da aplicação estar a ser corrida remotamente, algumas funcionalidades podem ser impossíveis de implementar totalmente sem recorrer a soluções adicionais (ex: integração com hardware externo).
- *Backup* e Migração de dados – Alguns prestadores de serviço na *Cloud* podem não oferecer funcionalidades de migração de dados ou políticas de backup que permitam facilmente transferir os dados de um prestador para outro. Para além disso também não existem garantias que após terminar o contrato com um determinado prestador de serviços, todos os nossos dados sejam devidamente destruídos.

Tendo em conta esta informação, e considerando que numa primeira fase a empresa pretende direccionar o produto SINGER para pequenos estabelecimentos, ficou praticamente claro que o modelo que mais se indicava ao nosso cenário seria a instalação no local. Para além disso, de acordo com os testemunhos recolhidos junto de agentes com experiência no terreno, existem actualmente no mercado nacional ainda muitos clientes (e potenciais novos clientes) sem acesso à Internet ou com serviços muito limitados (sem cobertura, velocidades baixas, limites de tráfego de dados, encargos extra para o cliente). Logo aqui é montada uma barreira à adopção do modelo em *Cloud* no nosso contexto.

Outro problema relevante que surge com a adopção do modelo na *Cloud* prende-se com o facto de ser necessário à empresa subscrever com entidades externas para fornecer os serviços na *Cloud*, pois neste momento esta ainda não tem dimensão suficiente para acomodar este tipo de serviços. Por outras palavras, isto significa que informação sensível dos clientes e do seu negócio passaria a estar visível a terceiros.

No entanto existe a perfeita noção que o futuro será dominado por serviços remotos na *Cloud* e por essa razão numa segunda fase poderemos vir adaptar o nosso sistema de modo a comunicar remotamente com servidores na *Cloud* (para sincronizar dados ou executar funcionalidades adicionais). Mas, para já, essa não será uma prioridade.

### 3. PROJECTO

Depois de uma primeira fase onde foi realizado um estudo de mercado ao nível de produtos concorrentes e das principais tecnologias disponíveis, procedeu-se ao levantamento de requisitos junto do cliente e à especificação da arquitectura do sistema SIGER.

Convém esclarecer que o sistema SIGER, tal como foi inicialmente pensado, inclui um vasto conjunto de módulos que serão desenvolvidos ao longo de um período de 2 anos. Porém, e dado que o estágio académico é realizado num período de tempo mais reduzido, o foco desse trabalho residiu fundamentalmente sobre os módulos Ponto de Venda e *Drink-Dispensers*. Isto não quer dizer que o trabalho realizado pelo estagiário se tenha restringido apenas a esses módulos, bem pelo contrário, o estagiário foi igualmente incentivado a participar na concepção e desenvolvimento dos restantes elementos. Contudo o foco deste relatório irá residir essencialmente sobre os módulos Ponto de Venda e *Drink-Dispensers*.

#### 3.1. ANÁLISE DE REQUISITOS

O processo de levantamento de requisitos consistiu num conjunto de etapas bem definidas que permitiram chegar a um entendimento mais concreto das necessidades dos utilizadores e dos objectivos do sistema. Passo a descrever o processo:

- a. Reconhecimento e avaliação do problema
- b. Especificar Requisitos
- c. Revisão junto do cliente

O ponto ‘a’ consistiu no primeiro contacto do analista com o cliente por forma a perceber as suas dificuldades e necessidades e a partir daí compreender a melhor forma de atacar o problema apresentado. Convém referir que o estagiário não teve participação directa nesta primeira fase, mas esteve presente em todas as outras etapas.

O ponto ‘b’ implicou a descrição detalhada de cada uma das funcionalidades pedidas pelo cliente. Relativamente aos requisitos funcionais, foi especificado uma descrição do requisito, as pré-condições e pós-condições esperadas, a prioridade do requisito e ainda um diagrama com o caso de uso relacionado. No fim foi igualmente gerado uma matriz de prioridades dos requisitos de maneira a filtrar as funcionalidades essenciais das que ficam para uma segunda abordagem. Esta filtragem teve de ser realizada devido aos constrangimentos temporais existentes para a conclusão da primeira fase do projecto.

Por fim temos a etapa ‘c’ que consistiu em reuniões presenciais com o cliente de forma a validar o trabalho realizado no ponto ‘b’ e garantir que não existiam inconsistências ou omissões relevante e que todos tinham a mesma visão do que o produto é suposto fazer.

Os pontos ‘b’ e ‘c’ foram executados num processo iterativo até se chegar a uma versão base do documento de levantamento de requisitos que satisfizesse o cliente e a equipa de desenvolvimento. Convém igualmente fazer referência ao facto que por tratar-se de um sistema criado de raiz, o levantamento de requisitos não teve de respeitar implementações ou abordagens seguidas previamente.



Dado a vasta quantidade de requisitos levantados, apenas irão ser enumerados os requisitos mais importantes, separados por tipo (funcionais e não funcionais):

### Requisitos Funcionais:

Abrir Sessão da Loja (Dia)	Este caso de uso permite que um utilizador com privilégios inicie automaticamente uma sessão para a loja no arranque do sistema. Na gíria, a sessão da loja corresponde ao período que vai desde o abrir do dia até ao fechar do dia.
Abrir Sessão do Empregado	Este caso de uso permite que um funcionário inicie a sua sessão no sistema. Antes de se poder iniciar uma sessão de empregado, o sistema já tem de possuir uma sessão de loja aberta e pelo menos uma sessão de caixa a decorrer.
Criar família de produtos	Este caso de uso permite que um utilizador com privilégios possa adicionar uma nova família de produtos ao sistema. Uma família de produtos consiste muito simplesmente num identificador escolhido pelo utilizador para representar um grupo de artigos semelhantes registados na base de dados do sistema.
Registar Pedido	Este caso de uso permite que um utilizador com privilégios adicione itens de venda a uma determinada venda activa. Esta venda activa contém todos os registos de artigos que foram pedidos pelo cliente e que até ao momento ainda não foram facturados.
Registar Pagamento	Este caso de uso permite que um utilizador com privilégios finalize uma venda activa e realize o recebimento do montante em dívida pelo cliente. Este recebimento pode ser feito de forma total ou parcial, de acordo com vários métodos de pagamento – venda a dinheiro – ou ainda transferindo o montante para uma conta corrente do cliente, momento em que a respectiva factura deve ser emitida. O sistema deve ainda permitir que a venda activa seja dividida em várias contas (Multi-Conta) e que estas sejam fechadas individualmente, ou que várias contas diferentes sejam pagas ao mesmo tempo emitindo apenas 1 documento fiscal. No fim, o sistema deve gerar uma factura que pode ou não ser impressa. O sistema deve contemplar também a situação em que o estabelecimento permite deixar contas activas entre aberturas e fechos de dia (hotéis por exemplo)
Registo das leituras dos dispositivos DDs	Ao servir uma bebida através de um equipamento <i>Drink-Dispenser</i> o sistema deve efectuar automaticamente o registo da quantidade de bebida servida, quando foi servida, por quem foi servida e creditar o total a facturar na conta do cliente servido. Esse registo deverá igualmente estar acessível para consulta posterior. Para além disso é igualmente esperado que os equipamentos <i>Drink-Dispenser</i> funcionem em modo de crédito/débito, ou seja, o serviço de uma determinada bebida só pode acontecer após o empregado realizar o registo do pedido do cliente no terminal POS.

Analisar consumos por bebidas	Esta funcionalidade pretende permitir consultar em tempo real os níveis de vendas (totais monetários ou volumes líquidos) de todas as bebidas servidas no estabelecimento e compará-las com os valores registados nos terminais POS de forma a identificar discrepâncias ou desvios invulgares.
-------------------------------	---

### Requisitos Não-Funcionais:

Categoria - Software	Por uma questão de redução de custos no preço final do produto e de forma a ter uma vantagem económica em relação à concorrência, o sistema será desenvolvido para um sistema Linux. A linguagem de programação será JAVA, o que nos permitirá num futuro e caso seja necessário, portar facilmente o produto para outros sistemas operativos. Pelas mesmas razões do ponto anterior, a escolha do SGBD irá recair sobre um produto <i>open-source</i> mas fiável e com uma grande aceitação no mercado actual. Para além disso o produto escolhido deverá seguir o modelo relacional.
Categoria - Segurança	Todas as operações executadas no sistema são associadas a um utilizador registado, e essa associação é realizada com base na autenticação/identificação do utilizador.
Categoria - Usabilidade	De forma a proporcionar a internacionalização do produto, este deverá suportar um conjunto de línguas diferentes, sem que isso afecte as funcionalidades que é capaz de desempenhar. Numa primeira fase o sistema terá de suportar pelo menos as seguintes línguas: <ul style="list-style-type: none"> <li>• PT-Português</li> <li>• EN-Inglês</li> </ul> No entanto no futuro está previsto que mais línguas venham a ser suportadas. Isto significa que irá existir um mecanismo de tradução de toda a UI (botões, texto, tabelas, etc.) que permita realizar a troca de língua de forma rápida e transparente para o utilizador, ao mesmo tempo que é possível actualizar o dicionário da aplicação sem ter de alterar e recompilar o código da aplicação.

Em **Anexo A** é incluído um excerto do documento de requisitos levantados e que serviu de base ao sistema que será implementado. Nesse documento são listados os requisitos funcionais e não funcionais, bem como a matriz de prioridade definida e validada junto do cliente.

## 3.2. ESPECIFICAÇÃO FUNCIONAL

A seguir ao processo de levantamento de requisitos passou-se à fase da especificação funcional do sistema SIGER. Esta etapa permitiu complementar o processo de levantamento de requisitos e dar ao cliente uma ideia visual do produto final. No entanto esta especificação foi realizada de forma geral e abrangente, ou seja, em vez de detalhar cada componente à exaustão, preferimos construir uma visão geral e integrada do que era esperado do produto final e assumindo sempre que esta viria a sofrer modificações no desenrolar do projecto.

Convém relembrar que o sistema SIGER originalmente projectado é constituído por vários módulos, no entanto, o foco da proposta de estágio restringe-se aos módulos do Ponto de Venda e dos *Drink-Dispensers*, e é sobre esses que irei continuar a dissecar neste relatório.

Este processo seguiu uma abordagem semelhante ao processo de levantamento de requisitos, onde em períodos de tempo fixos (semanalmente ou quinzenalmente) foram programadas reuniões como cliente de forma a validar o conteúdo produzido e obter esclarecimentos necessários. O objectivo principal da especificação funcional foi responder às seguintes questões:

- Identificar o fluxo de utilização do sistema mais adequado para este tipo de negócio, isto é, o SIGER está projectado para ser usado em ambiente *touch* e em cenários de utilização concorrida, logo tem de respeitar certos princípios de utilização.
- Apresentar de forma visual a implementação pensada para cada um dos requisitos para perceber se as expectativas do cliente foram correctamente interpretadas.
- Permitir não só à equipa adquirir um conhecimento do negócio como preparar as bases que permitissem a integração posterior de todos os módulos

A especificação funcional do módulo Ponto de Venda concebida seguiu uma abordagem tipo Manual Técnico do Utilizador, com a apresentação dos *Mockups* da UI da aplicação e listagem e descrição das funcionalidades oferecidas. Em **Anexo B** é disponibilizado um excerto da especificação funcional concebida para os módulos do Ponto de Venda e dos *Drink-Dispensers*.

Para melhor ilustrar o trabalho desenvolvido nesta etapa, incluo a seguir imagens *mockup* de algumas funcionalidades que fizeram parte da especificação inicial do projecto SIGER. Tal como é visível nas imagens a seguir, este interface foi posteriormente melhorado de forma a oferecer uma experiência de utilização mais agradável e consistente, algo que já tinha sido contemplado inicialmente.

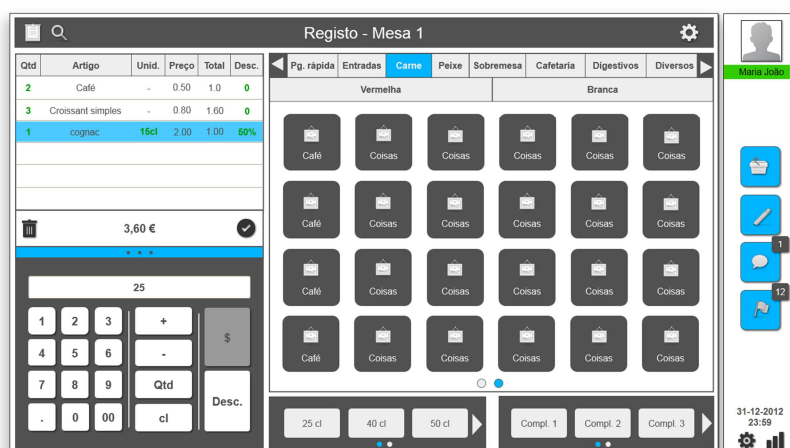


Figura 4 – Área de Registo (especificação inicial)

Esta janela corresponde ao ecrã de registo e consulta de vendas activas. Será a partir desta janela que os empregados poderão realizar o registo dos pedidos feitos pelos clientes e passar a seguir para a fase de pagamento e facturação final.

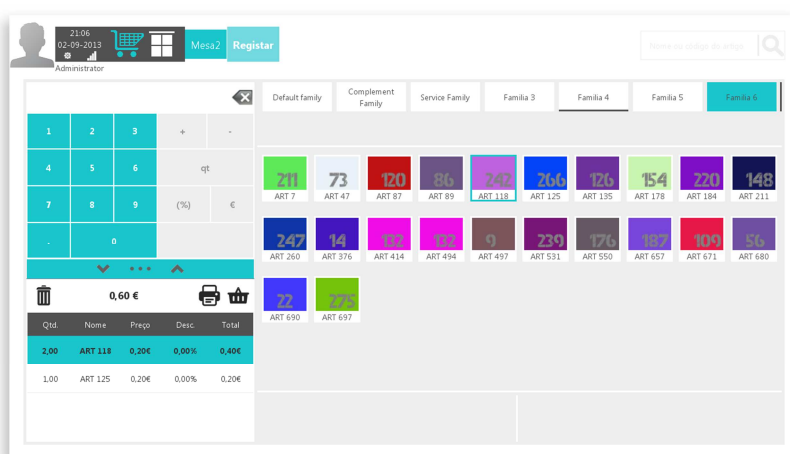


Figura 5 – Área de Registo (produto final)

As principais modificações visíveis nestas duas janelas prendem-se com a barra de menu principal que passou do lado direito para o topo da janela, a adopção de um *design* mais simplista baseado sobretudo em linhas rectas e ainda o uso de cores mais claras (o esquema de cores é personalizável).

### 3.3. ARQUITECTURA

O passo seguinte foi definir claramente a arquitectura geral do sistema SIGER. O objectivo nesta fase foi perceber como são integrados os vários componentes da aplicação de forma a construir um todo coeso e ao mesmo tempo manter um certo nível de abstracção na sua integração. Para melhor passar a ideia do modo como todo o sistema irá funcionar, optou-se por dividir a arquitectura do sistema em duas camadas, as quais irei apresentar em detalhe a seguir.

### 3.3.1. ARQUITECTURA FUNCIONAL

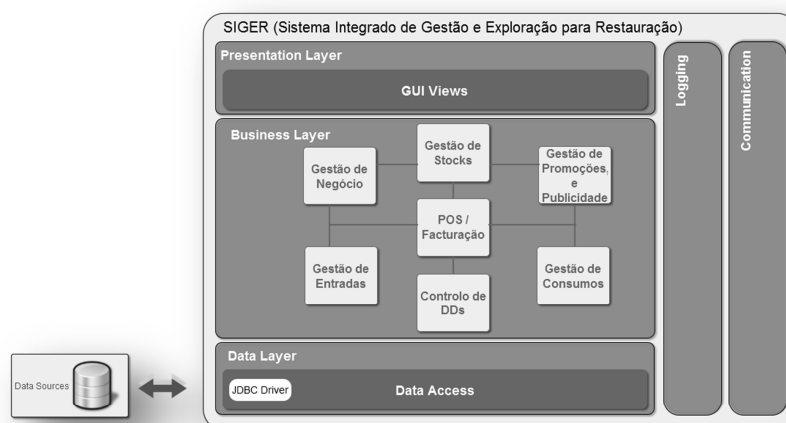


Figura 6 - Arquitectura Funcional

#### SIGER

<b><i>Business Layer</i></b>	Toda a lógica de trabalho do sistema que irá correr nos terminais POS está incluída neste módulo. É a partir daqui que as interações entre as várias camadas da aplicação serão orquestradas. No âmbito do estágio académico, é suposto o estagiário focar-se no desenvolvimento dos módulos POS/Facturação e de Controlo de <i>Drink-Dispensers</i> . De acordo com as suas especificidades, alguns módulos terão dependências de outros, tal como pode ser visto na Figura 6.
<b><i>Presentation Layer</i></b> → GUI VIEWS	A camada de apresentação inclui o conjunto de vistas da aplicação cliente que será a porta de entrada para o sistema. O objectivo com esta camada é separar a lógica de implementação, do ambiente gráfico disponibilizado aos utilizadores.
<b><i>Data Layer</i></b> → DATA ACCESS	Todo o sistema irá assentar sobre um modelo de dados relacional. Esta camada terá por finalidade garantir a comunicação com o motor de base de dados recorrendo a um <i>driver</i> JDBC de forma a garantir a integridade e salvaguardada de toda a informação transaccionada no sistema SIGER.
<b><i>Logging</i></b>	De forma a garantir um controlo mais rigoroso de todas as operações realizadas durante a execução da aplicação estas serão guardadas para log de modo a despistar possíveis falhas.

<b>Communication</b>	Este módulo será responsável por garantir a comunicação com o servidor de Base de dados e com os outros módulos que compõem o sistema SIGER. Para além disso existe ainda a necessidade de integrar os terminais POS com vários tipos de dispositivos externos (impressoras, balanças, leitores de código de barras/RFID/NFC, terminais X64, etc.) que será possível fazer através desta camada.
----------------------	--

## MÓDULOS

Tal como é visível na Figura 6, o sistema SIGER possui no seu core, um conjunto de funcionalidades oferecidas através da integração dos diferentes módulos. Passo a seguir a descrever sucintamente os vários módulos e principais funcionalidades oferecidas [1]:

<b>Módulo de Entradas / Saídas</b>	Este módulo é suposto oferecer funcionalidades de controlo e gestão das entradas/saídas dos clientes de um determinado estabelecimento. Este grupo de funcionalidades está mais direccionado para estabelecimentos nocturnos que necessitem de realizar este tipo de controlo. Este módulo não irá fazer parte dos planos de trabalho para este estágio.
<b>Módulo de Gestão de Negócio</b>	Este módulo permitirá fazer uma gestão simplificada de todo o negócio, fornecendo alguns indicadores de gestão e permitindo a análise estatística dos dados recolhidos. Pretende-se que o módulo seja simples de usar e que forneça indicadores práticos para uma gestão mais rentável. Este módulo não irá fazer parte dos planos de trabalho para este estágio.
<b>Módulo de Gestão de Stocks</b>	O problema da gestão de <i>Stocks</i> é que se trata de um processo moroso. Para evitar essa dificuldade pretende-se que o sistema de <i>Stocks</i> use intensivamente etiquetas RDID de modo a identificar inequivocamente os produtos e permitir uma gestão rápida e simplificada. Este módulo não irá fazer parte dos planos de trabalho para este estágio.
<b>Módulo de POS/Facturação</b>	Permitirá gerir as vendas de produtos aos clientes. Este módulo irá assegurar que todas as operações realizadas dentro de um determinado estabelecimento sejam correctamente transaccionadas e que os documentos emitidos sejam tratados de acordo com as regras de facturação em vigor. Este módulo incluirá igualmente as funcionalidades de <i>backoffice</i> necessárias para a configuração do sistema de acordo com a especificidade do negócio. Este módulo faz parte dos planos de trabalho para este estágio.
<b>Módulo de Promoções, Bolsa e Publicidade</b>	Permitirá a disponibilização aos clientes de informação de promoções, <i>Happy-Hours</i> , e outro tipo de publicidade. Este módulo contará igualmente

	<p>com uma ‘bolsa’ automática (segundo um algoritmo configurável), que ajustará o preço de acordo com a procura/oferta específica, dentro de limites pré-estabelecidos e podendo ter ou não em conta os <i>stocks</i> existentes. O objectivo desta bolsa é criar alguma dinâmica no consumo, ajudar a escoar <i>stocks</i> acumulados e permitir aumentar as vendas. Este módulo não irá fazer parte dos planos de trabalho para este estágio.</p>
<p><b>Módulo de Controlo de Drink-Dispensers</b></p>	<p>O módulo de controlo dos <i>DDs</i> irá permitir configurar e comunicar com os vários dispositivos de controlo de bebidas de modo a manter um controlo mais exacto dos valores realmente servidos dentro do estabelecimento. Este módulo faz parte dos planos de trabalho para este estágio.</p>
<p><b>Módulo de Gestão de Consumos</b></p>	<p>Tal como no módulo ‘Gestão de Entradas/Saídas’, também este foi pensado principalmente para os estabelecimentos nocturnos. Permite que cada cliente possa aceder aos seus consumos e obter sugestões personalizadas. Esta informação será disponibilizada em ecrãs distribuídos pelo estabelecimento. Este módulo não irá fazer parte dos planos de trabalho para este estágio.</p>

### 3.3.2. ARQUITECTURA TÉCNICA

A Figura 7 representa a arquitectura técnica do sistema, onde são evidenciados com mais detalhe todos os seus componentes e interacções.

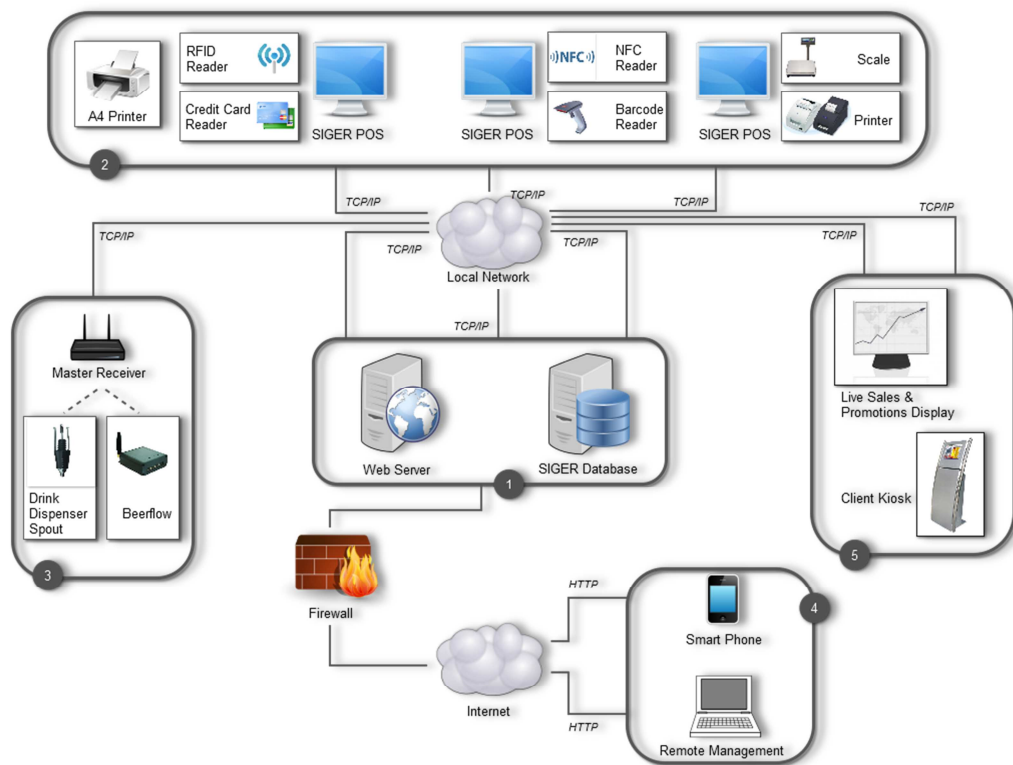


Figura 7 - Arquitectura Técnica

O sistema SÍGER irá assentar no modelo cliente/servidor, onde os vários intervenientes irão comunicar entre si através de uma rede TCP/IP. Para facilitar a leitura do diagrama foram agrupados em grupos os principais componentes do nosso sistema, que passo a descrever a seguir:

1. No grupo 1 são incluídos os diferentes servidores que fazem parte do ecossistema SÍGER. O sistema irá contar com um servidor de base de dados baseado em *Postgres*, por forma a manter o registo transaccional de todas as operações realizadas dentro da aplicação e necessárias ao negócio, e ainda um servidor HTTP para ter acesso remoto ao SÍGER. Na maioria dos cenários previstos, os vários servidores estarão instalados na mesma máquina física, podendo essa máquina em alguns casos, até correr inclusivamente uma versão da aplicação cliente (grupo 2).
2. No grupo 2 estão agrupados os clientes POS que serão a base do sistema e que terão por finalidade não só comunicar com o servidor mas também com um vasto conjunto de dispositivos auxiliares ao negócio. Entre os dispositivos a considerar temos balanças, impressoras A4, impressoras de talões, leitores de códigos de barra, leitores de RFID, leitores NFC, terminais rádio X64 e máquinas dispensadoras automáticas.
3. No grupo 3, estão esboçados os dispositivos que farão parte do módulo de controlo de bebidas e integrados no restante sistema. Os dispositivos a ter em conta são mais precisamente os ‘*Drink-Dispensers Spouts*’ e o ‘*Beer Control Flow*’. O primeiro dispositivo permite monitorizar os serviços de uma determinada garrafa ao ser acoplado no gargalo da mesma, enquanto o segundo monitoriza os fluxos de líquidos que passam através de um determinado canal. O *Master-Receiver* terá por finalidade integrar esses dispositivos comunicando com eles via radiofrequência.
4. É possível aceder ao sistema remotamente através de um acesso Web (HTTP/HTTPS). É neste cenário que entra em cena o servidor Web presente no grupo 1. Para salvaguardar a rede local de ataques provenientes de fora, está planeado incorporar uma *Firewall* para prevenir acessos indesejados.
5. Neste grupo juntamos os vários terminais que irão buscar a mais variada informação ao servidor para apresentar aos clientes do estabelecimento. Iremos ter um terminal com um display de dimensões consideráveis para apresentar as variações de preços dos artigos vendidos no estabelecimento ao longo do tempo e ao mesmo tempo estatísticas informativas acerca de alguns indicadores relevantes (ex. artigos mais vendidos). Por outro lado existirão igualmente terminais vocacionados para os clientes poderem consultar o estado das suas contas e consumos realizados.

### 3.4. MODELO DE DADOS

A próxima fase da especificação funcional passou por construir o modelo de dados que serve de suporte ao sistema SÍGER. Após a realização dos levantamentos de requisitos e de se ter uma ideia clara das funcionalidades a implementar nos vários módulos do projecto, foi possível identificar mais concretamente o tipo de informação persistente necessária ao correcto funcionamento da aplicação bem como o tipo de mapeamento entre classes e estruturas de dados necessárias.



Durante a fase de desenho da aplicação ficou definido *à priori* que nenhum registo da base de dados pode ser eliminado, isto é, quando o utilizador marcar um registo como apagado, este continua a existir na base de dados no entanto não voltará a ser apresentado ao utilizador. Contudo, rapidamente percebemos que esta estratégia não seria a mais indicada devido a questões de optimização do espaço disponível em disco e como tal esta abordagem foi esquecida.

Dado a sua grande dimensão não será possível apresentar aqui na sua totalidade o modelo de dados do sistema SIGER, no entanto, passo a seguir a descrever sucintamente alguns fragmentos de acordo com as funcionalidades previstas:

### ❖ Itens de Venda

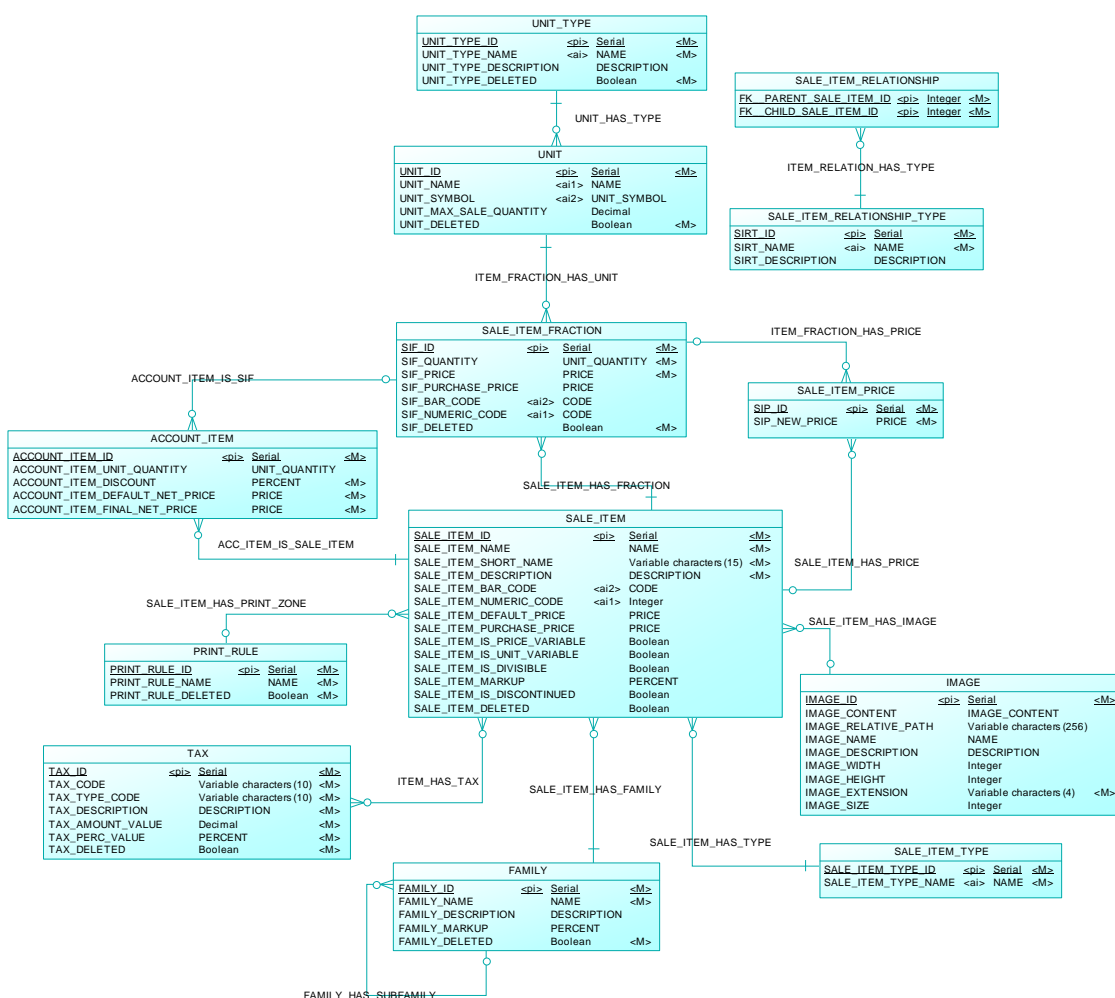


Figura 8 - Conjunto de entidades e relacionamentos para gestão de itens de venda

Este conjunto de entidades e relações representam os itens de venda transaccionados num qualquer negócio. Os itens podem estar agrupados por famílias e podem estar associados a uma zona de impressão. O nosso modelo permite diferenciar entre itens genéricos e subcategorias ou parcelas de itens.

❖ Grupos de utilizadores e Permissões

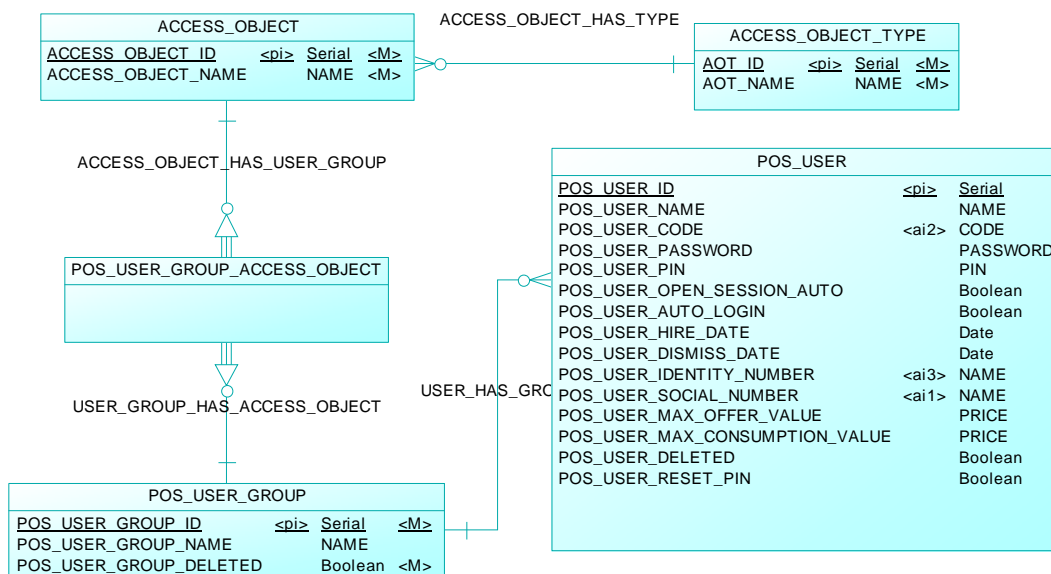


Figura 9 - Conjunto de entidades e relacionamentos para grupos e permissões

Este conjunto de entidades representa as relações existentes entre os utilizadores do sistema e os grupos de utilizadores criados. Os grupos de utilizadores possuem um conjunto de permissões associadas e que definem se os utilizadores têm acesso ou não a determinados conteúdos do sistema.

❖ Sessões de loja, caixa e utilizadores

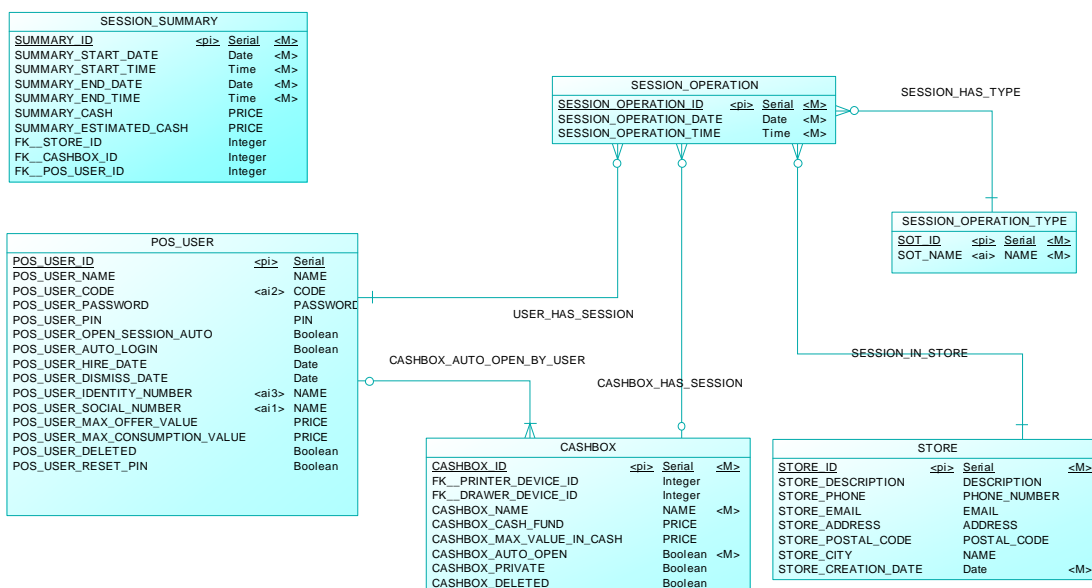


Figura 10 - Conjunto de entidades e relacionamentos para gestão de sessões

O sistema deve manter controlar o histórico de sessões iniciadas no sistema e para isso foram produzidas as seguintes entidades e relações. No fundo é necessário controlar as sessões para os utilizadores, caixas e lojas. A tabela ‘SESSION\_SUMMARY’ contém o histórico de todas as sessões realizadas no sistema (início e fim), enquanto a tabela ‘SESSION\_OPERATION’ apenas mantêm o registo das operações individuais realizadas.

❖ Documentos fiscais emitidos

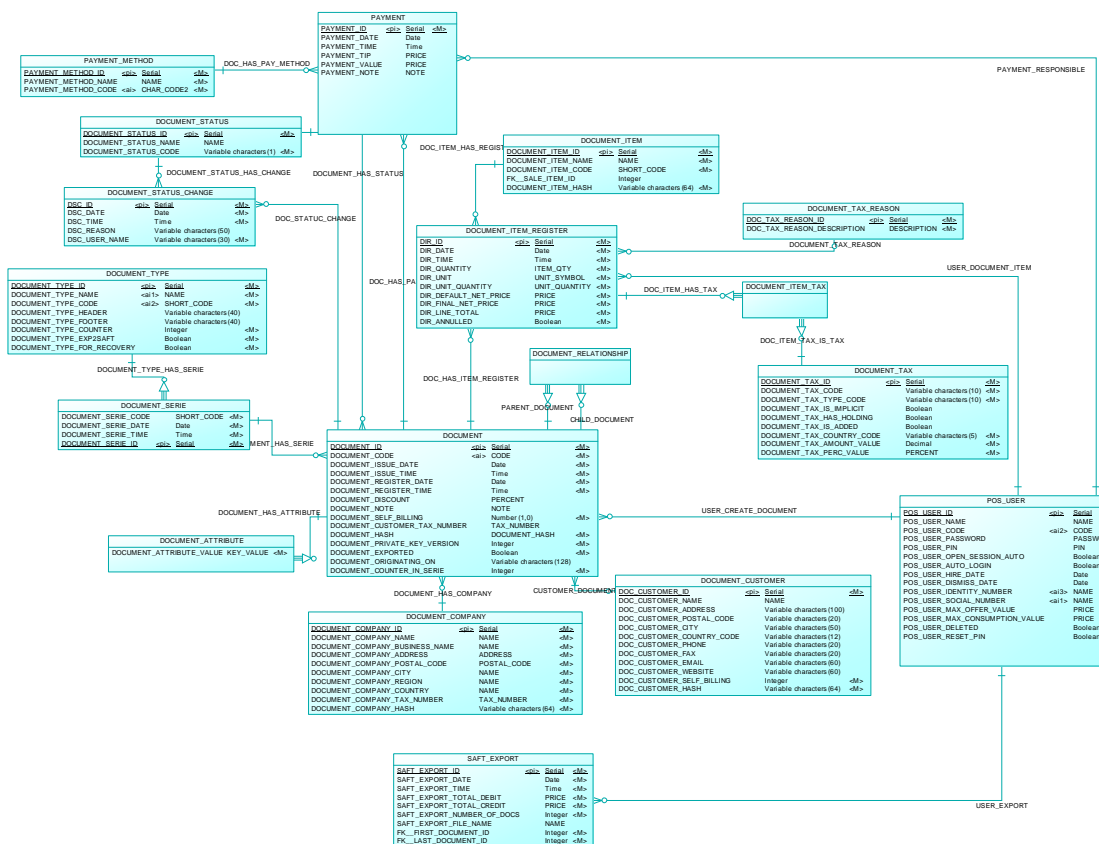


Figura 11 - Conjunto de entidades e relacionamento para gestão de documentos

Esta é possivelmente a parte mais importante do módulo POS, e que irá ser a base para produzir os relatórios SAF-T (PT) exigidos pelas Finanças. Toda a informação presente nestas entidades tem de ser única e não pode sofrer alterações ao longo do tempo. Para conseguir isso, sempre que existem dados que tenham de constar num determinado documento, estes são copiados integralmente das tabelas de origem e replicados para as entidades presentes na Figura 11. Os ficheiros SAF-T (PT) consistem na exportação integral da informação presente na base de dados relativo a um determinado intervalo de tempo seguindo um formato pré-estabelecido pela Autoridade Tributária e Aduaneira Portuguesa. A funcionalidade de exportação dos ficheiros SAF-T (PT) é um dos requisitos necessários para a certificação de qualquer tipo de Software de facturação (Portaria n.º 363/2010, de 23 de Junho [19] e Portaria n.º 22-A/2012, de 24 de Janeiro [20]).

## ❖ Gestão dos *Drink-Dispensers*

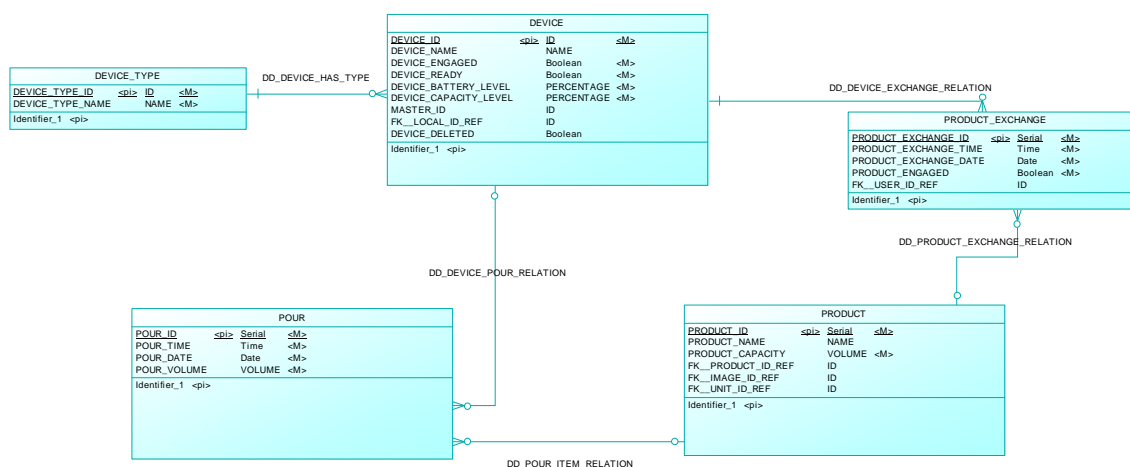


Figura 12 - Conjunto de entidades e relacionamentos para controle de DD

O modelo de dados para a gestão dos dispositivos de controlo de bebidas resume-se a estas poucas tabelas que permitem entre outros, manter a informação do estado e configuração de cada dispositivo, o histórico dos serviços realizados e ainda o histórico das trocas de bebidas efectuadas. Este modelo está projectado para poder ser utilizado como aplicação independente do SIGER (por exemplo para ser usado por aplicações de terceiros) e por isso não tem relações directas com as restantes tabelas do modelo de dados do SIGER. No entanto foram adicionados campos específicos de modo a funcionarem como chaves estrangeiras para tabelas de referência caso seja preciso (por exemplo: FK\_\_LOCAL\_ID para identificar o local onde se encontra o dispositivo).

### 3.5. TECNOLOGIAS

O projecto SIGER consiste num conjunto de módulos aplicativos a serem desenvolvidos de raiz e que por essa razão não exigem a necessidade de cuidados especiais ao nível de compatibilidade com funcionalidades ou tecnologias anteriores. Os principais requisitos aplicam-se sobretudo na necessidade da solução a desenvolver estar preparada para ser usada em ambientes *touch* e para correr em múltiplas plataformas. Aliado a essas exigências e devido a limitações de tempo para concluir o projecto, era importante que a escolha das ferramentas de desenvolvimento recaísse sobre tecnologias de alto nível e que permitissem uma prototipagem rápida.

Após um breve período de análise ao nível de tecnologias disponíveis, a escolha recaiu sobre o JavaFX [21]. O JavaFx pretende ser a evolução natural da plataforma Java para o desenvolvimento de aplicação cliente em modo *stand-alone* ou web e já com funcionalidades *multi-touch* de raiz. No entanto, após a criação da primeira prova de conceito baseado em JavaFX, deparámo-nos com problemas de performance significativas que não podiam ser negligenciadas. Mesmo incluindo optimizações o

desempenho continuava a ressentir-se em terminais com recursos mais limitados. Estes problemas obrigaram-nos a repensar a nossa escolha, que recaiu na adopção da *framework* Java Swing.

Na Figura 13, estão listadas todas as tecnologias e ferramentas que foram escolhidas para a concepção do projecto final.

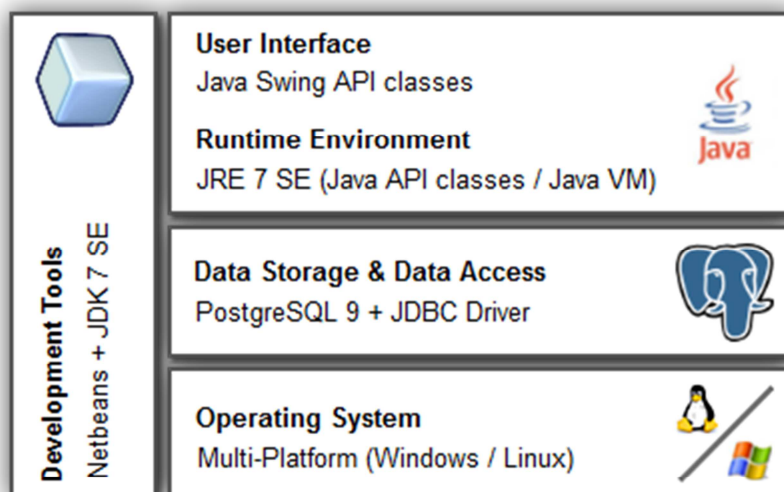


Figura 13 - Tecnologias e ferramentas de desenvolvimento

Nesta primeira fase é pretendido que a aplicação possa ser executada pelo menos em ambiente Windows e Linux. Para além disso, a adopção da *framework* Java permite-nos ter uma API homogénea de desenvolvimento, para as camadas de apresentação, core e de acesso a dados, num só ‘pacote’, sem necessidades de grandes configurações ou instalações complicadas.

Para a camada de persistência e acesso a dados ficou definido desde início que seria necessário optar por uma solução *open-source* por razões de custo do produto final. No entanto era necessário garantir a escolha de um produto com elevados padrões de qualidade, visto os dados transaccionados serem de extrema importância para os clientes e igualmente para os serviços fiscais. Por todas essas razões a escolha acabou por recair sobre o sistema PostgreSQL [22]. A comunicação entre a aplicação e o motor de base de dados ficou ao encargo dos drivers JDBC para PostgreSQL.

### 3.6. FERRAMENTAS E BIBLIOTECAS DE APOIO

Para a fase de desenvolvimento deste projecto foi necessário proceder ainda à instalação e configuração de algum *Hardware* e *Software* de apoio. Para começar, foi instalada uma máquina dedicada baseada em Linux dentro da rede local da empresa. Este servidor teve como finalidade controlar as várias versões do código fonte e documentação produzida (SVN) e ainda alojar uma plataforma de gestão de bugs de modo a termos um melhor acompanhamento do ciclo de vida dos vários problemas encontrados. A escolha da plataforma de gestão de bugs recaiu sobre o *Mantis Bug Tracker* [23].

De forma a garantir alguma segurança em caso de falhas de *Hardware*, foi necessário ainda criar um conjunto de *scripts bash* no servidor Linux que, de forma periódica, geram uma cópia de segurança, encriptam e sincronizam esses dados com um servidor remoto.

No que diz respeito à parte da programação, tivemos a necessidade de integrar algumas bibliotecas externas no nosso projecto de modo a dar resposta a algumas necessidades encontradas. Dada a falta de um suporte nativo por parte do JDK na comunicação I/O com as portas serial/paralelas, fomos obrigados a recorrer à biblioteca RxTx [24] de modo a conseguirmos uma abstracção com o sistema operativo e conseguirmos comunicar com os dispositivos externos via porta série/paralela. Uma grande vantagem desta biblioteca é o facto de permitir a sua utilização em vários sistemas operativos o que permite garantir a portabilidade da nossa aplicação para as várias plataformas.

Durante a integração dos dispositivos ‘*Drink-Dispensers*’ com a nossa aplicação surgiu ainda a necessidade de recorrer a mais uma biblioteca externa para gerir ligações HTTP. O fabricante disponibiliza actualmente uma interface Web para comunicação com o Master que de certa forma pretende ser uma alternativa mais aberta à biblioteca Skyman.dll (dependente da plataforma .Net). Para facilitar a gestão das comunicações e troca de pedidos HTTP com o servidor Web do *Master-Receiver*, optou-se por recorrer à biblioteca HttpComponents [25].

### 3.7. TESTES DE EXPLORAÇÃO REALIZADOS

Este capítulo visa apresentar os resultados testes obtidos sobre o sistema de controlo de bebidas constituído por um ou vários “*Master-Receiver*” e um ou vários “*Drink-Dispenser*”. Os dispositivos Skyflo [26] utilizados foram desenvolvidos pela empresa sediada em Hong-Kong “*Sky Channel Technologies*”. Mais concretamente, existiu a possibilidade de contactar com os seguintes dispositivos:

- “*Master-Receiver*” : Unidade de controlo de toda a infraestrutura de “*Drink-Dispensers*” emparelhados nela. Existe igualmente a possibilidade de encadear “*Master-Receiver*” entre si de forma a criar uma rede de controlo mais complexa.
- “*DD Skyflo Pour Spout*”: Dispositivo que é encaixado no topo da garrafa (tipo rolha) e tem como função medir com precisão a quantidade de líquido que é servido e reportar esses valores ao *Master-Receiver*.



Figura 14 - Dispositivo *Master-Receiver*

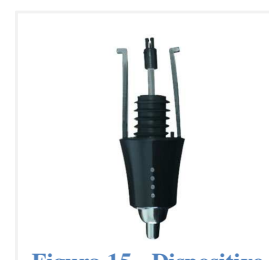


Figura 15 - Dispositivo *Pour Spout*

Este tipo de dispositivo é capaz de funcionar em 2 modos diferentes:

- **Modo 1 - “Normal”:** Neste modo o dispositivo limita-se a enviar os valores das doses servidas para o seu MR. O líquido só é servido quando a garrafa ao qual o dispositivo está conectado, atingir um certo angulo de inclinação (parâmetro configurável).
- **Modo 2 - “Crédito/Débito”:** Neste modo, o dispositivo entra em modo de bloqueio e não deixa servir líquidos até que seja enviado um comando de crédito para o seu MR. O comando de crédito permite várias combinações de parâmetros por forma a precisar com exactidão quem deve servir o quê e em que quantidade/doses.
- **“DD Optic-Pearl”:** Dispositivo semelhante ao anterior mas mais limitado em termos de funcionalidades. A função principal é a mesma, ou seja, contabilizar a quantidade de líquido que é servida, porém neste caso, o dispositivo já vem com tamanhos fixos de fábrica (por ex. 25ml). Isto quer dizer, que cada uma das leituras enviadas para o MR corresponde sempre a uma dose inteira correspondente ao tamanho do dispositivo. Para além disso, este dispositivo não é capaz de trabalhar no modo 2.



Figura 16 - Dispositivo Optic Pearl

A ligação ao *Master-Receiver* pode ser realizada via Ethernet ou por porta serial (interface RS.232). Os registos de doses servidas são lidas directamente da memória interna do Master e mediante a API de comunicação com é possível navegar pelos registos que estão aí guardados.

Existem 3 tipos de pacotes de leitura (xml) que podem ser lidos do Master através da API skyman. Estes são:

- *Pour Data*
- *Info Data*
- *Empty Data*

A seguir é apresentado o exemplo de um pacote “*Pour Data*”:

```
<?xml version="1.0" encoding="utf-8"?>
<Protocol>
  <MID Data="1639445297" />
  <CMD Data="145" />
  <Option Data="65535" />
  <Voat Data="25" />
  <Date Data="2012/08/10" />
  <Time Data="14:37:05" />
  <DID Data="1407881524" />
  <PDN Data="0" />
```



```

<Waiter Data="0" />
<Table Data="12" />
<Station Data="0" />
<Status Data="2" />
<Channel Data="0" />
<Dfcs Data="12" />
</Protocol>

```

Estes são os pacotes que contêm a informação da quantidade de líquido servido, a data, hora, o ID do DD e outros parâmetros importantes.

Exemplo de um pacote “*Info Data*”:

```

<?xml version="1.0" encoding="utf-8"?>
<Protocol>
  <MID Data="1639445297" />
  <CMD Data="146" />
  <Option Data="65535" />
  <ExStatus Data="64" />
  <Date Data="2012/08/10" />
  <Time Data="14:08:33" />
  <DID Data="679214484" />
  <PDN Data="0" />
  <Waiter Data="0" />
  <Table Data="1" />
  <Station Data="0" />
  <Status Data="131" />
  <Channel Data="0" />
  <Dfcs Data="1" />
</Protocol>

```

Este tipo de pacotes servem para sinalizar notificações gerados pelos dispositivos sobre o estado de algum tipo de evento ocorrido. Estes eventos podem ser por exemplo, falta de bateria ou ainda a sinalização de troca de garrafa, ou outro tipo de informação útil.

Exemplo de um pacote “*Empty Data*”:

```

<?xml version="1.0" encoding="utf-8"?>
<Protocol>
  <MID Data="1639445297" />
  <CMD Data="147" />
</Protocol>

```

Os pacotes “*Empty data*” não possuem qualquer tipo de informação útil, são usados apenas para avisar que já não existem mais registos de leituras para ler.

Como já foi mencionado anteriormente, os dispositivos usados nestes testes permitem operar em 2 modos diferentes (simples e com créditos/débitos). Para alterar o modo de funcionamento foi necessário proceder à configuração do dispositivo pretendido e que consistiu em enviar ao *Master-Receiver* um pacote XML com o conjunto de novos valores para os parâmetros RfTask e RfSkip. A partir desse momento os dispositivos deixam de servir livremente e só permitem servir após ser reencaminhado um comando de crédito para o *Master-Receiver*.

Exemplo de um comando de crédito:

```
<?xml version="1.0" encoding="UTF-8"?>
<Protocol>
<MID Data="1639445297" />
<CMD Data="33"/>
<Option Data="36864"/>
<Voat Data="134" />
<DID Data="" />
<PDN Data="" />
<Waiter Data="" />
<Table Data="" />
<Station Data="" />
<Channel Data="" />
<DType Data="" />
<Overtime Data="" />
<SerialNum Data="0" />
</Protocol>
```

Este tipo de comando permite, entre outros, identificar claramente o dispositivo alvo (DID), o produto que deve ser servido (PDN), o empregado (Waiter), a mesa (Table) e o volume permitido (Voat).

Em certa altura, surgiu a necessidade proceder a uma actualização do *Master-Receiver* para podermos despistar alguns problemas que foram surgindo. Para tal procedeu-se à actualização da *firmware* do dispositivo e à reposição de um conjunto de ficheiros necessários na sua memória interna.

A forma oferecida pelo fabricante para integrar com estes dispositivos consiste na utilização de uma biblioteca externa baseada em .NET ou então via a troca de pedidos POST sob o protocolo HTTP. A biblioteca Skyman.dll requer que a aplicação esteja a correr num ambiente com *framework* .NET instalada, o que acaba por impossibilitar a portabilidade directa entre várias plataformas (Windows, Linux e Mac). Por outro lado, apesar da integração por HTTP ser independente do sistema operativo em causa, existe o problema de este protocolo não suportar todas as funcionalidades oferecidas pela biblioteca Skyman.dll original. A principal falha nesta solução prende-se sobretudo com a impossibilidade de controlar os dispositivos em modo de crédito/débito.

## 4. IMPLEMENTAÇÃO

Após finalizada a fase de especificação do projecto, deu-se então início às tarefas de implementação do produto final. Este capítulo tem por objectivo apresentar o trabalho final produzido e ainda expor resumidamente os principais desafios encontrados.

### 4.1. TRABALHO DESENVOLVIDO

O trabalho realizado ao longo deste estágio académico resultou na produção de uma aplicação cliente final que engloba o conjunto de requisitos funcionais base definidos no início do processo de implementação. Nesta primeira versão da aplicação foram incluídas funcionalidades de registo e facturação de vendas, bem como as funcionalidades de *backoffice* dos dados necessários ao seu correcto funcionamento. Esta aplicação foi desenhada para correr em terminais equipados com interface *touch*, e em modo de execução exclusiva, isto é, ao iniciar a máquina e após carregar o sistema operativo, a aplicação é lançada em modo *fullscreen*. Por sua vez, ao sair da aplicação, a máquina automaticamente trata de limpar os recursos usados e encerrar o sistema.

Neste capítulo irei apresentar com base em imagens reais retiradas durante uma utilização normal da aplicação, algumas funcionalidades desenvolvidas.

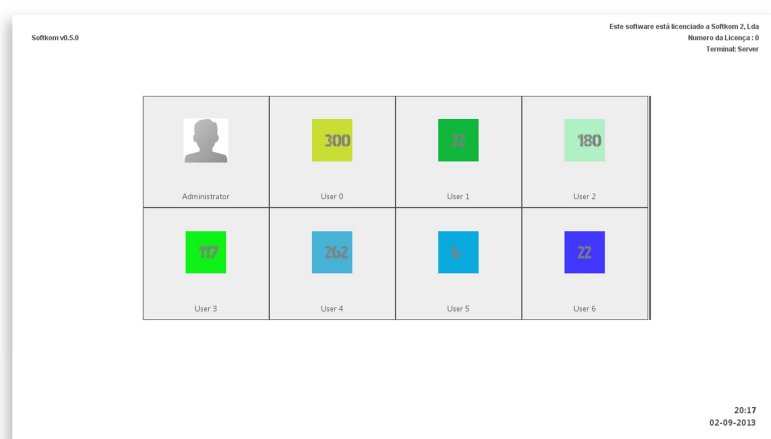


Figura 17 - Selecção de Perfis & Login

Ao iniciar a aplicação, o utilizador é confrontado com um ecrã de selecção de perfis (Figura 17) e só após validar o seu PIN secreto, é que é dado acesso ao sistema (de acordo com as permissões associadas). O controlo de acessos à aplicação não foi só uma forma natural encontrada para prevenir acessos indesejados ao sistema mas é igualmente um requisito imposto pela Autoridade Tributária para poder passar a certificação obrigatória neste tipo de aplicações.

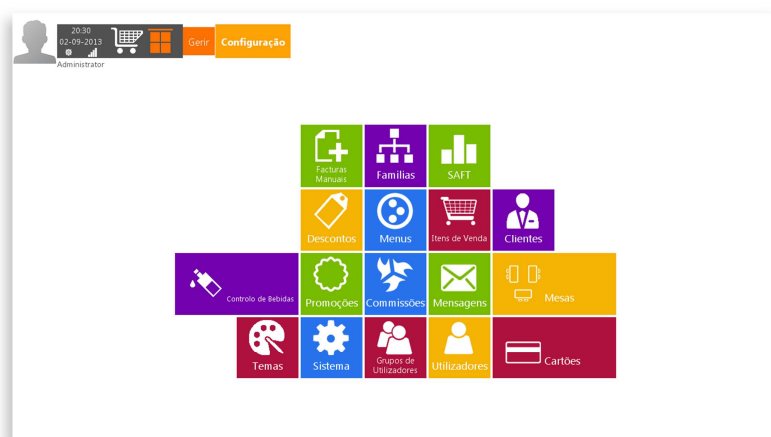


Figura 18 - Menu Configuração

Após obter acesso ao sistema o utilizador pode navegar (de acordo com as suas permissões) através dos vários menus do sistema. À medida que o utilizador vai progredindo na hierarquia de menus, vai sendo apresentado no topo do ecrã o caminho percorrido, através do qual o utilizador pode sempre voltar atrás. Na barra de navegação de topo são igualmente fornecidas opções rápidas para aceder à área de registo, área de gestão ou ainda para fazer *logout* e bloquear o sistema (seleccionando a imagem de perfil).

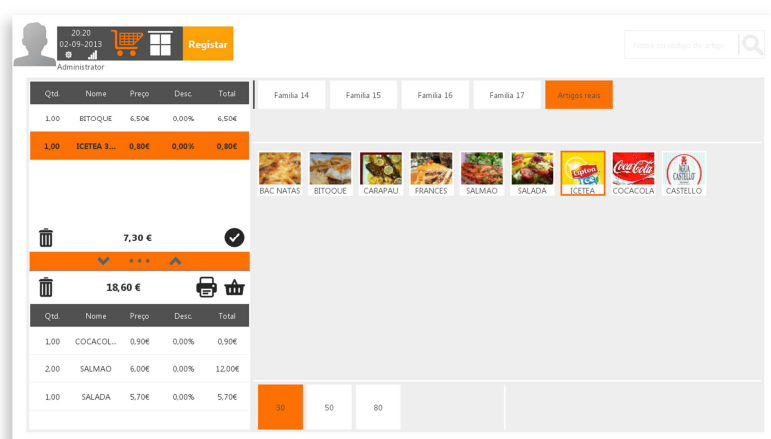


Figura 19 - Registo de Vendas

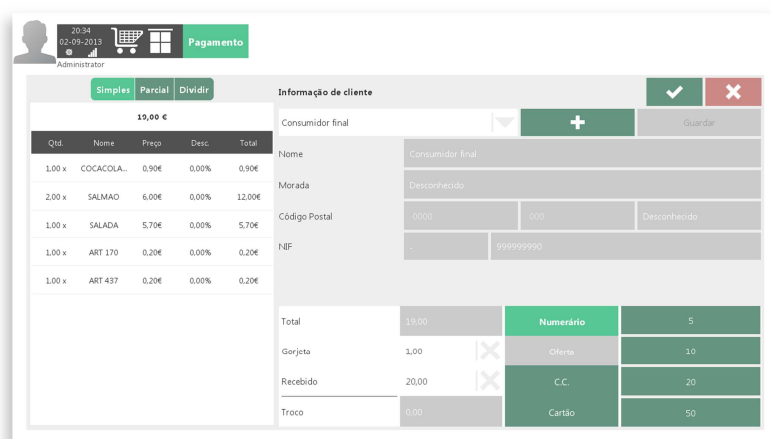


Figura 20 - Finalizar Venda [Pagamento]

As principais funcionalidades oferecidas pelo módulo de facturação incluem o registo de vendas e pagamento/emissão de documentos fiscais. Estas funcionalidades estão disponíveis nos ecrãs representados na Figura 19 e Figura 20 respectivamente. Na janela de registo de vendas o utilizador pode navegar pelas famílias de artigos ou procurar directamente o artigo pretendido, adicionar/remover artigos das listas, alterar quantidades, preços, descontos e unidades (quando aplicável). Após finalizar o processo de registo e no momento de fechar a conta, o utilizador é encaminhado para a janela apresentada na Figura 20. Aqui o utilizador tem a possibilidade de efectuar vários tipos de pagamento (simples, parcial ou a dividir), escolher o cliente final, adicionar clientes passantes, escolher o método de pagamento e o montante a registar. No final, e dependendo do tipo de configuração presente no terminal pode ser apresentado visualmente no ecrã o documento emitido (Figura 21) ou ainda ser impresso nas respectivas impressoras seleccionadas. A impressão dos documentos é feita baseada em regras configuráveis que definem com base no local de origem e o tipo de documento, as impressoras de destino.

1 / 1

Softkom 2, Lda  
Leiria  
3000  
123456789

-----  
FAC-882013/1

Data: 2013-09-02  
Hora: 20:35  
Terminal: Server  
Empregado: Administrator

-----

Qt Prod	IVA	SubTot
1,00 COCCOOLA33cl	-	0,50
2,00 SALMIO	-	12,00
1,00 SALADA	-	5,70
1,00 ART 170	0%	0,20
1,00 ART 437	0%	0,20
		-----
Taxa	Base	Imposto
0%	0,40	0,00
		Total
		0,40

-----

Total € : 19,00

-----

Pagamento :

2013-09-02 20:35 Dinheiro 19,00

-----

Nome: Consumidor final  
NIF: 999999999  
Morada: Desconhecido  
CPostal: Desconhecido

-----

d/96 - processado por programa certificado  
99

Figura 21 - Impressora de ecrã

Convém realçar que a aplicação foi igualmente preparada para suportar a mudança de esquemas de cores em *runtime*. Os esquemas de cores são carregados dinamicamente no arranque da aplicação a partir de ficheiros de configuração personalizáveis e durante a execução da aplicação, o utilizador pode em qualquer altura alternar entre os esquemas existentes. Para além da personalização ao nível das cores é também possível escolher de entre um conjunto de tamanhos de letras disponíveis. Esta personalização foi pensada sobretudo para facilitar na acessibilidade a clientes com maiores dificuldades de visão. A variação do esquema de cores é visível no conjunto de figuras incluídas neste capítulo.

Outra funcionalidade não menos importante consiste na procura, consulta e reimpressão de documentos previamente emitidos no sistema. É exactamente com essa finalidade que foi criada a janela de consulta de documentos, aqui representada pela Figura 22. Nesta área o utilizador pode realizar pesquisas de acordo com vários parâmetros, consultar rapidamente os itens que pertencem ao documento ou ainda efectuar determinadas operações especiais sobre o documento seleccionado (por exemplo: reimprimir, realizar reembolso, etc.).

The screenshot shows the 'Consultar Documentos' (Document Search) interface. At the top, there's a header with the user 'Administrator' and the date '02-09-2013'. Below this are search filters: 'Empregado' (Employee), 'Data início' (Start date) with a calendar icon, 'Data fim' (End date) with a calendar icon, 'Locais' (Locations), 'Documento' (Document), and 'Item de venda' (Sales item). There are also fields for 'N.º de documento' (Document number) and 'NIF' (Tax ID). Buttons for 'Limpar pesquisa' (Clear search) and 'Procurar' (Search) are present.

Below the filters is the 'Dados gerais' (General data) section, which contains a table with the following data:

Número	Total	Cliente	Data de emissão	Qtd	Item	Unidade	Preço	Total
FAC-522013/1	19,00 €	Consumidor final [999999990]	02-09-2013 20:35	2,00	Salmao	-	6,00 €	12,00 €
				1,00	Salada	-	5,70 €	5,70 €
				1,00	Artide ...	-	0,20 €	0,20 €
				1,00	Artide 4..	-	0,20 €	0,20 €

At the bottom of the table, there is an 'Acções' (Actions) dropdown menu.

Figura 22 - Consulta de Documentos

Os sistemas de facturação que sejam comercializados hoje em dia, necessitam de passar um processo de certificação junto da Autoridade Tributária de modo a averiguar que este está a trabalhar de acordo com a legislação fiscal em vigor. Um dos requisitos para poder passar essa certificação, é a necessidade do sistema ser capaz de produzir um ficheiro SAF-T válido. O SAF-T (PT) (*Standard Audit File for Tax Purposes – Portuguese version*) é um ficheiro normalizado (em formato XML) com o objectivo de permitir uma exportação fácil, e em qualquer altura, de um conjunto predefinido de registos contabilísticos, de facturação ou de documentos de transporte, num formato legível e comum, independentemente do programa utilizado, sem afectar a estrutura interna da base de dados do programa ou a sua funcionalidade. Na Figura 23 é apresentado o ecrã com as funcionalidades de exportação do ficheiro SAF-T. Esta janela permite igualmente consultar rapidamente o histórico de exportações realizadas até ao momento.

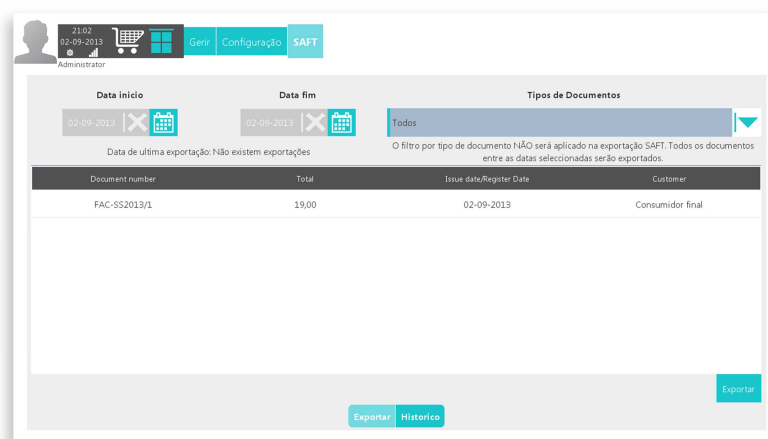


Figura 23 - Exportação de ficheiros SAF-T

Para além das funcionalidades de facturação, gestão de documentos e exportação SAFT implementadas, foram ainda produzidas as funcionalidades de *backoffice* necessárias para a configuração do sistema de modo a adaptá-lo às necessidades do negócio dos vários clientes. Passo a citar de forma resumida, as principais funcionalidades de *backoffice* implementadas:

- Adicionar/Editar/Eliminar famílias de artigos (as famílias permitem agrupar de forma lógica um conjunto de artigos semelhantes)
- Adicionar/Editar/Eliminar artigos
- Adicionar/Editar/Eliminar complementos (complementos são artigos especiais que permitem personalizar o produto base no momento do pedido)
- Modificar preferências regionais tais como moeda ou formato decimal
- Editar propriedades dos tipos de documentos
- Criar novas séries de documentos (os documentos fiscais necessitam de ser todos assinados com uma série única)
- Adicionar/Editar/Eliminar utilizadores
- Adicionar/Editar/Eliminar grupos de utilizadores (os grupos de utilizadores representam um conjunto de actores com permissões de acesso semelhantes)
- Editar permissões de acesso
- Adicionar/Editar/Eliminar locais (um local corresponde a um nome lógico que o utilizador usa para classificar uma determinada zona do seu estabelecimento)
- Adicionar/Editar/Eliminar caixas (as caixas são entidade lógicas para representar os diferentes destinos físicos para o dinheiro em numerário, tais como as gavetas e as bolsas)
- Adicionar/Editar/Eliminar clientes
- Adicionar/Editar/Eliminar regras de impressão
- Adicionar/Editar/Eliminar mesas
- Adicionar/Editar/Eliminar dispositivos
- Abrir/Fechar Sessões de utilizadores, caixa e loja

Como já foi dito anteriormente, o sistema foi pensado para ser usado em ambientes *touch*, no entanto a JDK do Java não oferece actualmente soluções para auxiliar na introdução de texto dessa forma. Por essa razão foi necessário desenvolver de raiz uma

biblioteca auxiliar para o teclado virtual. Este teclado é apresentado sempre que algum tipo de componente de introdução de texto é seleccionado e escondido no momento em que é perdido o *focus*. Na Figura 24 é visível a aplicação do teclado produzido.

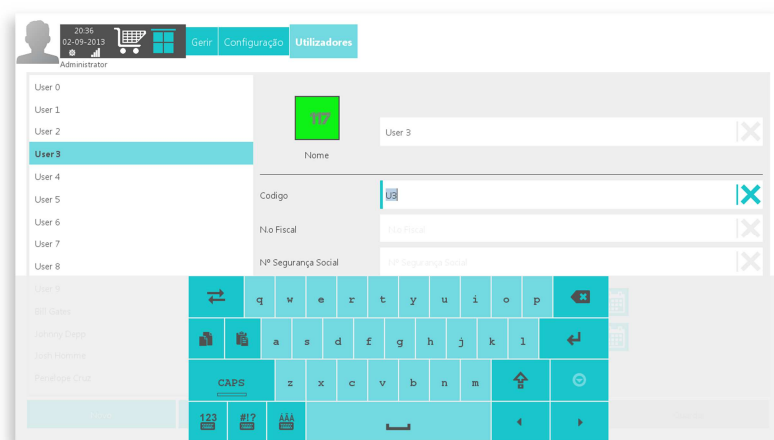


Figura 24 - Gestão de Utilizadores

Por fim falta apenas introduzir as funcionalidades desenvolvidas para a gestão dos dispositivos de controlo de bebidas (*Drink-Dispensers*). Estas operações pertencem a um módulo auxiliar ao módulo de facturação e exigem a integração com dispositivos especiais apresentados mais atrás neste documento. A integração entre a nossa aplicação e os dispositivos disponíveis foi conseguida mediante a troca de pedidos HTTP com o controlador *Master-Receiver*.

Resumidamente através da janela presente na Figura 25, o utilizador pode controlar os dispositivos instalados, adicionar/remover dispositivos novos, controlar as várias trocas de garrafas realizadas por dispositivo e ainda controlar os totais facturados vs os totais realmente servidos. Esta informação é fundamental para despistar desperdícios e otimizar as vendas através de um controlo mais rigoroso.

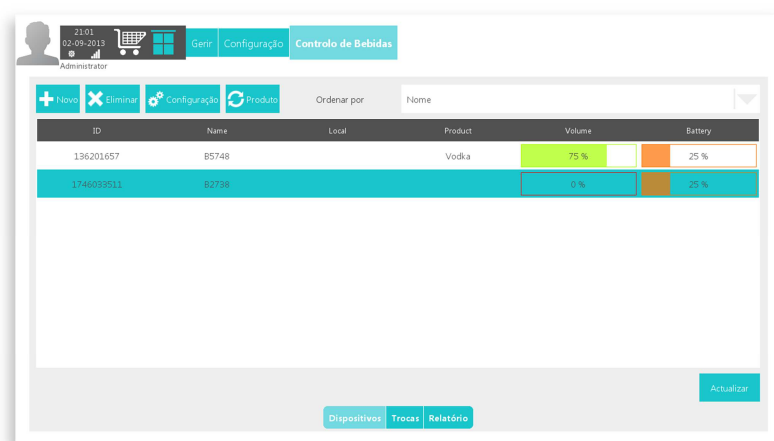


Figura 25 - Controlo de DDs



## 4.2. DESAFIOS ENCONTRADOS

Ao longo do desenrolar do projecto foram surgindo vários desafios, que irei tentar analisar sucintamente neste capítulo. O objectivo aqui é apresentar não só os problemas detectados mas também as respectivas soluções encontradas.

### 4.2.1. MODELO DE DADOS

A fase de análise e concepção do modelo de dados que permitiu dar suporte ao projecto SIGER acabou por se revelar mais desafiante do que era esperado inicialmente. O facto de o projecto estar pensado para se adaptar a vários modelos de negócio diferentes, tais como por exemplo restaurantes, lojas de roupas, tabacarias, cafés, etc. e dada a sua complexidade em termos de variáveis envolvidas, isto obrigou a que houvesse um planeamento inicial mais cuidadoso. Mesmo tendo em conta esta fase inicial de análise do problema, já durante a fase de desenvolvimento foi necessário por momentos operar algumas alterações de modo a corrigir algumas falhas introduzidas inicialmente ou mesmo para responder a necessidades que foram surgindo no desenrolar da fase de implementação, nomeadamente as várias alterações exigidas por questões de exportação do SAF-T.

### 4.2.2. ESTRUTURAÇÃO DO CÓDIGO

O objectivo deste projecto visava desenvolver uma solução modular para facturação e apoio ao negócio de estabelecimentos na área da restauração/retalho. De modo a permitir abstrair a parte visual da aplicação do núcleo de funções implementadas optou-se por dividir o projecto em 2 partes distintas, o Core e a UI. Isto permite que no futuro se possa facilmente mudar a ‘cara’ da aplicação reaproveitando ao máximo as funções nucleares ou então ainda produzir um versão para *Android* mantendo sempre a estrutura base e sem necessidades de muitas alterações.

Ao nível da estruturação das classes do *core* foi ainda tida em conta a possibilidade de este projecto vir a ser integrado por terceiros e como tal tentou-se criar uma interface para as chamadas às funções nucleares do projecto. Essa interface consiste num conjunto de classes *Managers* que oferecem uma abstracção de alto nível para as rotinas implementadas no core, tal como está representado na Figura 26.

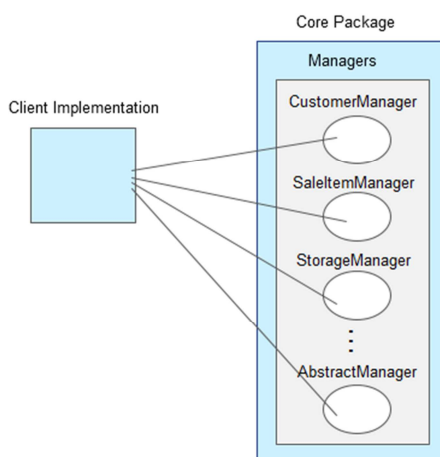


Figura 26 - Interface para o core da aplicação

### 4.2.3. COMUNICAÇÃO ESC/P (IMPRESSORAS)

Um dos requisitos básicos exigidos para o projecto SIGER, era que este permitisse realizar a impressão de documentos. Nesta primeira fase contudo, o essencial era que essa impressão fosse realizada em papel de 40 colunas (papel talão), em impressoras fornecidas pela empresa. Para controlar e comunicar com este tipo de impressoras optou-se por adoptar o standard ESC/P (Epson Standard Code for Printers). Este standard desenvolvido pela Epson (mas usado por diferentes marcas) exige que a comunicação com o dispositivo seja feita através de uma porta serial/paralela e mediante a troca de bits de dados seguindo um protocolo bem definido.

Mais tarde surgiu igualmente a necessidade de apresentar no ecrã (pré-visualização) uma cópia exacta dos documentos impressos, e por essa razão foi criada uma classe que conseguisse simular os vários comandos definidos no protocolo e realizasse as operações de impressão para o ecrã em vez do papel. Tanto a classe responsável por imprimir para ecrã como a classe responsável por imprimir para porta série estendem de uma interface comum com a definição das várias funções ESC/P suportadas.

Convém realçar ainda que visto que existiu a necessidade de comunicar com portas série e dada a falta de suporte na JDK base do Java para responder a esse problema, fomos obrigados a recorrer a uma biblioteca externa para colmatar essa falha. A nossa escolha acabou por recair na biblioteca RxTx [24] devido ao seu elevado grau de maturidade e por permitir o seu uso nas várias plataformas disponíveis (Windows, Linux, Mac).

### 4.2.4. INTEGRAÇÃO COM O *MASTER-RECEIVER*

Como já foi dito anteriormente, a forma *standard* oferecida pelo fabricante para comunicar com os dispositivos de controlo de bebidas (*Drink-Dispensers*) resumia-se a uma biblioteca Dll baseado na plataforma .Net. Esta dependência à plataforma .Net colidia com a nossa necessidade do sistema ser portátil para as várias plataformas (Windows, Linux, Mac) sem necessidade de grandes alterações. No entanto, após entrarmos em contacto com o fabricante, fomos informados que existiam umas actualizações ao *Hardware* e *Firmware* que permitiam um tipo de integração mais aberto e independente da plataforma base. Essa comunicação passaria a ser realizada via protocolo HTTP mediante o envio de comandos POST bem definidos. A resposta do *Master-Receiver* consiste num pacote resposta cujo conteúdo inclui um conjunto de dados segundo o protocolo definido pelo fabricante (em vários formatos CSV, XML, HEX).

Depois de ultrapassada esta primeira barreira e após termos acesso à documentação do fabricante com detalhes acerca do protocolo de comunicação definidos, seguiu-se um breve período de estudo e análise. Apesar de o protocolo não ser muito complicado de desenvolver a verdade é que a qualidade da documentação não era a melhor. Por várias vezes foram detectadas algumas inconsistências entre o formato de dados descritos na documentação e as respectivas respostas obtidas do *Master-Receiver*.

De qualquer forma foi possível finalmente integrar o nosso módulo de facturação directamente com os dispositivos *Drink-Dispensers* embora com algumas limitações. Ao princípio pretendia-se que o controlo dos dispositivos pudesse ser realizado em modo crédito/débito, isto é, os dispositivos só permitiam servir as quantidades pretendidas após envio dos pedidos para o Master, no entanto esse modo de utilização não é suportado através do protocolo HTTP. As razões que nos foram dadas pelo fabricante foi de que devido a limitações de Hardware no *Master-Receiver*, tornava-se difícil realizarem esse tipo de gestão. Por fim, falta só referir que de modo a abstrairmos a nossa implementação do protocolo HTTP, tivemos de recorrer à biblioteca *HttpComponents* da Apache [25].

#### 4.2.5. CERTIFICAÇÃO DA APLICAÇÃO

Por forma a poder legalmente ser lançada para o mercado nacional, a aplicação necessita de ser certificada pela Autoridade Tributária e Aduaneira. Para isso tem de respeitar um conjunto de parâmetros e regras de acordo com o lançado em diário da república. As principais regras a ter em conta prendem-se com a forma de assinar os documentos fiscais emitidos pelo sistema e ainda o formato e conteúdo dos dados a constar nos documentos SAF-T (PT) exportados.

A assinatura dos documentos fiscais emitidos (factura, factura simplificada, guia de remessa, etc.) obriga a que seja gerada uma assinatura através do algoritmo RSA com base num conjunto de campos presentes no documento e na chave privada do produtor do programa. Esta assinatura gerada deve ser guardada juntamente ao registo do documento em base de dados e é necessário ainda apresentar um subgrupo de 4 caracteres da chave, nos documentos impressos.

O SAF-T (PT) é um ficheiro que contém dados contabilísticos fiáveis que se podem exportar de um sistema contabilístico original por um período de tempo específico, e que se lê facilmente em virtude da sua standardização de *layout* e formato (XML), que pode ser usado pelos funcionários das autoridades tributárias, com o fim de verificar e validar os movimentos realizados por um determinado cliente. Este documento necessita de ser enviado periodicamente para os servidores da AT (Autoridade Tributária) e deverá sempre ser efectuada para um determinado exercício, contendo todos os documentos/transacções até à data de extracção.

O desafio maior neste capítulo residiu na complexidade do formato deste tipo de ficheiro e por vezes na compreensão do significado de alguns parâmetros. Para além disso, visto que o ano de 2013 foi um ano recheado de alterações ao nível das regras de facturação, por diversas vezes fomos obrigados a proceder a modificações consecutivas sobre trabalho realizado anteriormente. Estas alterações acabaram por introduzir alguns atrasos no plano de trabalho previsto.

#### 4.2.6. TESTES

De forma a validar a qualidade do trabalho produzido foi necessário submeter os vários módulos desenvolvidos a um conjunto de testes rigorosos. No entanto os testes realizados foram divididos em várias fases que passo a descrever a seguir:

## Testes unitários & integração

Por testes unitários classificam-se todos os testes realizados durante a fase de programação e que incidem directamente sobre as unidades de código produzidas no momento. Por unidade de código entende-se o fragmento mais pequeno de código que possa ser testado. O objectivo destes testes é averiguar que o código implementado produz as saídas esperadas.

Através dos testes de integração pretendeu-se assegurar que os vários módulos de código produzidos funcionavam de acordo com o que seria esperado quando integrados entre si. Neste cenário seguiu-se uma abordagem *bottom-up*, onde após realizar os testes unitários às várias unidades de código passamos a integração dos vários componentes de forma a validar os valores de saída e garantir o correcto funcionamento da interface entre componentes.

Tanto os testes unitários como a maioria dos testes de integração, por se tratarem principalmente de testes de caixa branca (*White Box Testing* – Testes que necessitam que se conheça os mecanismos internos do código em teste), ficaram à responsabilidade de cada programador nas unidades de código produzidas por si.

## Testes funcionais

Os testes funcionais têm como finalidade assegurar que os requisitos produzidos inicialmente estão correctamente implementados e passam os critérios de validação definidos. Este tipo de testes enquadra-se na categoria de testes de caixa preta (*Black Box Testing* – Testes que ignoram os mecanismos internos do sistema e focam-se apenas nos valores de saída produzidos). Nesta fase a equipa optou por adoptar um modelo de testes cruzados, onde as funcionalidades desenvolvidas por um programador seriam testadas por um outro programador que não tenha participado no desenvolvimento dessas mesmas funcionalidades.

## Testes de aceitação & beta

No fim da implementação dos módulos de Facturação e *Drink-Dispensers*, houve ainda lugar à sessão de testes de aceitação junto do cliente final. Esta fase envolveu a participação do CEO da empresa (cliente final) e de alguns colaboradores não técnicos da empresa de forma a assegurar que o produto final correspondia às expectativas traçadas durante o levantamento dos requisitos e especificação do projecto. Findo este período de testes junto do cliente foi igualmente disponibilizado uma versão de demonstração do produto final para ser instalada no terreno e assim realizar testes *beta* em cenários de utilização real.

## 5. CONCLUSÃO E PRÓXIMAS TAREFAS

### 5.1. CONCLUSÃO

Findo o período delineado para o estágio académico, chega a altura de fazer o balanço do trabalho realizado e das experiências retiradas ao longo deste ano. Apesar de tratar-se do primeiro contacto directo do estagiário com o mundo profissional, o apoio oferecido pelos restantes elementos da equipa permitiu uma integração rápida e fácil.

Desde logo, foi possível perceber que comparativamente ao ambiente académico, os imprevistos são frequentes e se estes não forem ajuizadas correctamente, podem ditar o fim prematuro de um projecto de *Software*. Isto foi principalmente visível neste projecto, no momento em que a equipa de desenvolvimento se viu na necessidade de alterar significativamente a especificação do projecto de forma a acomodar a implementação do módulo de facturação.

Tornou-se claro que o tempo investido inicialmente na especificação do produto e no estudo do mercado e formas de operar, permitiu iniciar a fase de desenvolvimento com mais confiança e desde logo eliminar um conjunto de incertezas que apareceram logo no início do projecto.

Em suma, as experiências retiradas ao longo deste ano permitiram ao estagiário desenvolver uma série de aptidões essenciais no mercado de trabalho tais como por exemplo o trabalho em equipa, a gestão correcta do tempo, o espírito de autonomia e iniciativa. Para além disso, esta experiência permitiu ainda ao estagiário contactar directamente com um mercado de trabalho que até aqui lhe era totalmente desconhecido (área do retalho/restauração).

### 5.2. TRABALHO FUTURO

Tal como pode ser visto na Figura 6, o projecto SIGER inclui um vasto conjunto de módulos que não são contemplados para o âmbito deste estágio por razões de limitação de tempo. Apesar desta fase do estágio não incluir esses módulos, algum trabalho já teve de ser feito de forma a acomodar os requisitos desses módulos, principalmente ao nível do desenho da base de dados e na especificação geral de como estes se integram entre si.

Para além disso, devido a alguns atrasos nos trabalhos por parte de equipas externas ou mesmo devido à necessidade de alterações radicais no plano de trabalhos da nossa equipa, algumas tarefas que envolviam a integração com tecnologias mais inovadoras tais como o RFID e NFC, tiveram de ser retiradas do plano do estágio e acabaram por sofrer um adiamento no tempo. De qualquer forma é perfeitamente claro que a introdução deste tipo de funcionalidades no nosso produto irá permitir uma diferenciação importante relativamente à concorrência.

Um aspecto negativo que foi possível de destacar ao longo deste trabalho e que deveria ser melhorado no futuro prende-se com a necessidade de incluir algum tipo de automatismos no processo de testes da aplicação. O facto de ter havido a necessidade de proceder a sucessivas alterações ao longo do tempo (devido por exemplo a alterações na

legislação), por vezes as novas alterações acabaram por introduzir problemas novos em zonas que já tinham sido testadas previamente e que obrigaram a uma nova ronda de testes e correcção de bugs. Se tivesse existido a possibilidade de periodicamente realizar uma serie de testes de regressão, isso teria permitido minimizar o impacto produzido novas alterações. Claro que a criação deste tipo de soluções obrigaria a investir mais tempo em tarefas complementares e o ideal seria encontrar aqui um equilíbrio ideal.

## REFERÊNCIAS

- [1] Softkom, “SIGER - Especificação Inicial do Produto,” 2011.
- [2] “Grupo PIE - A empresa,” [Online]. Available: <http://www.grupopie.com/empresa.html>. [Acedido em Dezembro 2012].
- [3] “WinRest Homepage,” [Online]. Available: <http://www.winrest.pt/>. [Acedido em Dezembro 2012].
- [4] “PingWin Homepage,” [Online]. Available: <http://www.pingwin.ws/>. [Acedido em Dezembro 2012].
- [5] “ZoneSoft - ZSRest,” [Online]. Available: <http://www.zonesoft.pt/site/category/zsrest>. [Acedido em Dezembro 2012].
- [6] “Resumo das principais funcionalidades do software ZSRest,” [Online]. Available: [http://www.zonesoft.pt/zsrest\\_stuff/zsrest.pdf](http://www.zonesoft.pt/zsrest_stuff/zsrest.pdf). [Acedido em Dezembro 2012].
- [7] “Sage GesRest II - A nova geração em software para a Restauração,” [Online]. Available: <http://www.sage.pt/Default.aspx?action=ArticleViewer&target=1935>. [Acedido em Dezembro 2012].
- [8] “Gestwin POS,” [Online]. Available: [http://www.softpack.pt/index.php/grupo\\_produtos/gestwin-pos-2/](http://www.softpack.pt/index.php/grupo_produtos/gestwin-pos-2/). [Acedido em Dezembro 2012].
- [9] “HarborTouch - Features,” [Online]. Available: <http://www.harbortouch.com/pos-solutions/hospitality/features/>. [Acedido em Dezembro 2012].
- [10] “TMN Wallet,” [Online]. Available: <http://wallet.tmn.pt/index.html>. [Acedido em ]  
] Dezembro 2012].
- [11] “Pssst! Primavera,” [Online]. Available:  
] <http://www.primaverabss.com/pt/OverviewPssst.aspx>. [Acedido em 06 2013].
- [12] “Square Register,” [Online]. Available: <https://squareup.com/register>. [Acedido em ]  
] 06 2013].
- [13] “Flat Design,” [Online]. Available: <http://james-mcdonough.com/flat-design-aesthetic/>. [Acedido em 06 2013].
- [14] “Radio-Electronics.com - RFID Radio Frequency Identification Technology Tutorial,” [Online]. Available: <http://www.radio-electronics.com/info/wireless/radio-frequency-identification-rfid/technology-tutorial-basics.php>. [Acedido em Dezembro 2012].
- [15] “What's the difference between RFID and NFC?,” [Online]. Available:  
] <http://electronics.howstuffworks.com/difference-between-rfid-and-nfc.htm>. [Acedido em Dezembro 2012].
- [16] “How Will NFC Impact the Hospitality Industry?,” [Online]. Available:  
] <http://www.askidentive.com/nfc-technology/nfc-impact-hospitality-industry>. [Acedido em Dezembro 2012].
- [17] “Impinj - Applications > RFID Case Study: Tavern Restaurant Group,” [Online].  
] Available:  
] [http://www.impinj.com/Documents/Applications/Case\\_Studies/Tavern\\_Restaurant\\_Group\\_Case\\_Study/](http://www.impinj.com/Documents/Applications/Case_Studies/Tavern_Restaurant_Group_Case_Study/). [Acedido em Dezembro 2012].
- [18] “Cloud, Pros & Cons,” [Online]. Available:  
] <http://www.javacodegeeks.com/2013/04/advantages-and-disadvantages-of-cloud-computing-cloud-computing-pros-and-cons.html>. [Acedido em 06 2013].

- [19 M. d. F. e. d. A. “Portaria n.º 363/2010, de 23 de Junho,” [Online]. Available:  
] [http://info.portaldasfinancas.gov.pt/NR/rdonlyres/DD5DE0DB-41F8-4DB3-8D12-5B6AE01425DD/0/Portaria\\_363\\_2010.pdf](http://info.portaldasfinancas.gov.pt/NR/rdonlyres/DD5DE0DB-41F8-4DB3-8D12-5B6AE01425DD/0/Portaria_363_2010.pdf).
- [20 M. d. F. e. d. A. “Portaria n.º 22-A/2012, de 24 de janeiro,” 24 Janeiro 2012.  
] [Online]. Available: <http://info.portaldasfinancas.gov.pt/NR/rdonlyres/8C23078D-3B1F-45EE-9D02-C01C2DCA6949/0/portaria22A2012.pdf>.
- [21 “JavaFX - The Rich Client Platform [Oracle],” [Online]. Available:  
] <http://www.oracle.com/technetwork/java/javafx/overview/index.html>. [Acedido em Dezembro 2012].
- [22 “Postgresql Homepage,” [Online]. Available: <http://www.postgresql.org/>. [Acedido em  
] em Dezembro 2012].
- [23 “Mantis Bug Tracker,” [Online]. Available: <http://www.mantisbt.org/>. [Acedido em  
] 06 2013].
- [24 “RxTx,” [Online]. Available: [http://rxtx.qbang.org/wiki/index.php/Main\\_Page](http://rxtx.qbang.org/wiki/index.php/Main_Page).  
] [Acedido em 06 2013].
- [25 “Apache HttpComponents,” [Online]. Available: <http://hc.apache.org/>. [Acedido em  
] 06 2013].
- [26 “The Skyflo wireless system,” [Online]. Available: <http://kipflo.com/home.asp>.  
] [Acedido em Dezembro 2012].



ANEXOS

ANEXO A

*Excerto do documento de levantamento de requisito inicial, produzido no âmbito do desenvolvimento do projecto SIGER.*

... (Página 56/94) ...

[RF001] ABRIR VENDA ACTIVA

<b>Este caso de uso permite que um utilizador com privilégios abra uma venda activa para que posteriormente seja possível adicionar itens a essa venda. Esta venda activa irá conter todos os registos de artigos que forem pedidos pelo cliente e que até ao momento ainda não foram facturados.</b>	
<b>Prioridade:</b>	Essencial
<b>Entrada e pré-condições:</b>	Entrada: Identificador único daquela venda activa para a qual serão associados os itens pedidos.
	Pré-Condição: O utilizador identifica-se e o sistema percebe que este tem os devidos privilégios associados e por isso apresenta-lhe o formulário para abrir a venda activa.
<b>Saída e pós-condições:</b>	Saída: ---
	Pós-Condições: A venda activa fica disponível para serem adicionados itens.
<b>Diagrama Caso de Uso</b>	<pre> graph TD     Actor[Employee with privileges] --- UC1((Manage Active Account))     UC2((Open Account)) -.-&gt; &lt;&lt;extend&gt;&gt;  UC1     UC3((Authentication)) -.-&gt; &lt;&lt;use&gt;&gt;  UC1     </pre>

[RF002] REGISTAR PEDIDO

<b>Este caso de uso permite que um utilizador com privilégios adicione itens de venda a uma determinada venda activa. Esta venda activa contém todos os registos de artigos que foram pedidos pelo cliente e que até ao momento ainda não foram facturados.</b>	
<b>Prioridade:</b>	Essencial
<b>Entrada e pré-condições:</b>	Entrada: Itens para adicionar e a venda activa onde devem ser incluídos.
	Pré-Condição: O utilizador identifica-se e o sistema percebe que este tem os devidos privilégios associados e por isso apresenta-lhe o formulário para escolher os itens a adicionar à venda activa.
<b>Saída e pós-condições:</b>	Saída: ---
	Pós-Condições: Os itens adicionados passarão a fazer parte da lista de itens da venda activa a facturar.
<b>Diagrama Caso de Uso</b>	<pre> graph TD     Actor[Employee with privileges] --- UC1((Manage Active Account))     UC2((Place Order)) -.-&gt; &lt;&lt;extend&gt;&gt;  UC1     UC3((Search Account)) -.-&gt; &lt;&lt;include&gt;&gt;  UC2     UC4((Authentication)) -.-&gt; &lt;&lt;use&gt;&gt;  UC1     </pre>

[RF003] ANULAR PEDIDO

<b>Este caso de uso permite que um utilizador com privilégios corrija uma conta activa aberta.</b>	
<b>Prioridade:</b>	Essencial
<b>Entrada e pré-condições:</b>	Entrada: A conta activa pretendida e as alterações a serem incutidas nessa conta.
	Pré-Condição: O utilizador identifica-se e o sistema percebe que este tem os devidos privilégios associados e por isso apresenta-lhe o formulário para corrigir os artigos que se encontram na conta activa.
<b>Saída e pós-condições:</b>	Saída: ---
	Pós-Condições: A conta activa associada irá a partir desse momento reflectir as alterações incutidas. Todas as alterações efectuadas pelo empregado serão registadas num histórico para permitir perceber se foram cometidas ilegalidades.
<b>Diagrama Caso de Uso</b>	<pre> graph TD     Actor[Employee with privileges] --- UC1((Manage Active Account))     UC1 -- use --&gt; UC2((Authentication))     UC1 -.-&gt; UC3((Cancel Order))     UC3 -.-&gt; UC1     UC3 -.-&gt; UC4((Search Account))     UC4 -.-&gt; UC1     UC4 -.-&gt; UC5((Issue Invoice))     UC5 -.-&gt; UC4     </pre>

[RF004] REGISTRAR PAGAMENTO

<b>Este caso de uso permite que um utilizador com privilégios finalize uma venda activa e realize o recebimento do montante em dívida pelo cliente. Este recebimento pode ser feito de forma total ou parcial, de acordo com vários métodos de pagamento – venda a dinheiro – ou ainda transferindo o montante para uma conta corrente do cliente, momento em que a respectiva factura deve ser emitida. O sistema deve ainda permitir que a venda activa seja dividida em várias contas (multi-conta) e que estas sejam fechadas individualmente, ou que várias contas diferentes sejam pagas ao mesmo tempo emitindo apenas 1 documento fiscal. No fim, o sistema deve gerar uma factura que pode ou não ser impressa. O sistema deve contemplar também a situação em que o estabelecimento permite deixar contas activas entre aberturas e fechos de dia (hotéis por exemplo).</b>	
<b>Prioridade:</b>	Essencial
<b>Entrada e pré-condições:</b>	Entrada: Conta activa e método de pagamento escolhido.
	Pré-Condição: O utilizador identifica-se e o sistema percebe que este tem os devidos privilégios associados e por isso apresenta-lhe o formulário para registar o pagamento do cliente.
<b>Saída e pós-condições:</b>	Saída: ---
	Pós-Condições: A venda activa será facturada e arquivada no sistema.
<b>Diagrama Caso de Uso</b>	<pre> graph TD     Actor[Employee with privileges] --- UC1((Manage Active Account))     UC1 -- use --&gt; UC2((Authentication))     UC1 -.-&gt; UC3((Pay/Close account))     UC3 -.-&gt; UC1     UC3 -.-&gt; UC4((Search Account))     UC4 -.-&gt; UC1     UC3 -.-&gt; UC5((Issue Invoice))     UC5 -.-&gt; UC3     </pre>

[RF005] REALIZAR ESTORNO

<b>Este caso de uso permite que um utilizador com privilégios invalide uma factura previamente processada mas ainda antes do fecho da caixa.</b>	
<b>Prioridade:</b>	Essencial
<b>Entrada e pré-</b>	Entrada: A factura referenciada.

<b>condições:</b>	Pré-Condição: O utilizador identifica-se e o sistema percebe que este tem os devidos privilégios associados e por isso apresenta-lhe o formulário para realizar o estorno de uma determinada factura registada no sistema e tendo em conta que a caixa ainda não foi fechada.
<b>Saída e pós-condições:</b>	Saída: --- Pós-Condições: É lançado pelo sistema uma factura com valor negativo que corrija a factura referenciada.
<b>Diagrama Caso de Uso</b>	<pre> graph TD     Actor[Employee with privileges] --- UC1((Manage Invoices))     UC1 -.-&gt; &lt;&lt;extend&gt;&gt;  UC2((Reverse Invoice))     UC1 -.-&gt; &lt;&lt;use&gt;&gt;  UC3((Authentication))     </pre> <p>The diagram shows an actor 'Employee with privileges' connected to a use case 'Manage Invoices'. 'Manage Invoices' has an 'extend' relationship with 'Reverse Invoice' and a 'use' relationship with 'Authentication'.</p>

[RF006] REALIZAR REEMBOLSO

**Este caso de uso permite que um utilizador com privilégios invalide uma factura previamente processada já depois do fecho da caixa.**

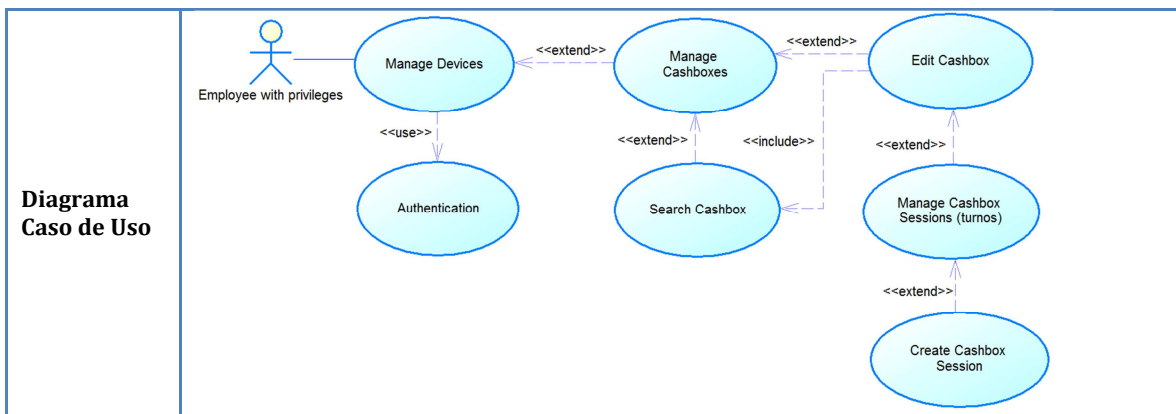
<b>Prioridade:</b>	Essencial
<b>Entrada e pré-condições:</b>	Entrada: A factura referenciada. Pré-Condição: O utilizador identifica-se e o sistema percebe que este tem os devidos privilégios associados e por isso apresenta-lhe o formulário para realizar o reembolso de uma determinada factura registada no sistema e tendo em conta que a caixa já foi fechada.
<b>Saída e pós-condições:</b>	Saída: --- Pós-Condições: É lançado pelo sistema uma factura com valor negativo que corrija a factura referenciada.
<b>Diagrama Caso de Uso</b>	<pre> graph TD     Actor[Employee with privileges] --- UC1((Manage Invoices))     UC1 -.-&gt; &lt;&lt;extend&gt;&gt;  UC2((Refund))     UC1 -.-&gt; &lt;&lt;use&gt;&gt;  UC3((Authentication))     </pre> <p>The diagram shows an actor 'Employee with privileges' connected to a use case 'Manage Invoices'. 'Manage Invoices' has an 'extend' relationship with 'Refund' and a 'use' relationship with 'Authentication'.</p>

Sessões

[RF007] FECHAR TURNO

**Este caso de uso permite que um utilizador com privilégios feche um turno que foi automaticamente aberto no momento de abertura de caixa. Este fecho é feito manualmente, onde o funcionário deve fazer a contagem do dinheiro e registar essa contagem no sistema.**

<b>Prioridade:</b>	Desejável
<b>Entrada e pré-condições:</b>	Entrada: Dados da novo turno a registar no sistema. Pré-Condição: O utilizador identifica-se e o sistema percebe que este tem os devidos privilégios associados e por isso apresenta-lhe o formulário para inserir os dados do novo turno a criar.
<b>Saída e pós-condições:</b>	Saída: O novo turno criado. Pós-Condições: O novo turno é registada no sistema e passa a ser utilizável pelos restantes módulos.



[RF008] INICIAR SESSÃO DA LOJA (DIA)

<b>Este caso de uso permite que um utilizador com privilégios inicie automaticamente uma sessão para a loja no arranque do sistema. Na gíria, a sessão da loja corresponde ao período que vai desde o abrir do dia até ao fechar do dia.</b>	
<b>Prioridade:</b>	Essencial
<b>Entrada e pré-condições:</b>	Entrada: Instante temporal em que é iniciada a sessão.
	Pré-Condição: O utilizador identifica-se e o sistema percebe que este tem os devidos privilégios associados e por isso apresenta-lhe o formulário para iniciar manualmente uma sessão da loja.
<b>Saída e pós-condições:</b>	Saída: ---
	Pós-Condições: A partir deste momento todas as caixas e respectivas sessões abertas serão associadas à sessão acabada de criar.
<b>Diagrama Caso de Uso</b>	

[RF009] FECHAR SESSÃO DA LOJA (DIA)

<b>Este caso de uso permite que um utilizador com privilégios finalize uma sessão para a loja. Na gíria, a sessão da loja corresponde ao período que vai desde o abrir do dia até ao fechar do dia. O fecho do dia pode ser feito de duas formas:</b>	
<b>Não cego:</b> implica que todas as vendas activas, sessões de funcionário e caixas estejam fechadas, seja feito um apuramento do dinheiro em gavetas e multibanco, seja registado o valor de fundo maneio para o dia seguinte e que seja impresso um relatório.	
<b>Cego:</b> as vendas activas, sessões de funcionário e caixas devem estar fechadas, deve ser registado o valor de fundo maneio para o dia seguinte, mas não é necessário fazer o apuramento de dinheiro nem imprimir qualquer relatório.	
<b>Prioridade:</b>	Essencial
<b>Entrada e pré-condições:</b>	Entrada: Instante temporal em que é fechada a sessão.
	Pré-Condição: O utilizador identifica-se e o sistema percebe que este tem os devidos privilégios associados e por isso apresenta-lhe o formulário para fechar manualmente uma sessão da loja.
<b>Saída e pós-condições:</b>	Saída: ---
	Pós-Condições: A partir deste momento todas as caixas e respectivas sessões abertas serão fechadas a não ser que existam vendas activas nalguma delas. Nesse caso deverá ser perguntado ao utilizador como é que ele quer tratar as vendas activas.
<b>Diagrama Caso de Uso</b>	

[RF010] ABRIR CAIXA

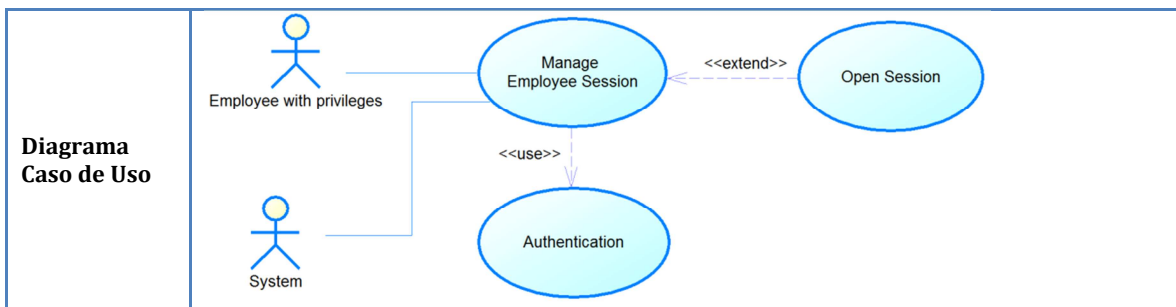
<b>Este caso de uso permite que um utilizador com privilégios abra manualmente uma caixa. O abrir de caixa exige que o utilizador defina o fundo de maneio disponível na caixa lógica. Por abrir de caixa entenda-se o acto de iniciar uma sessão para a caixa.</b>	
<b>Prioridade:</b>	Essencial
<b>Entrada e pré-condições:</b>	Entrada: Instante temporal em que é iniciada a sessão e o fundo de maneio dessa caixa.
	Pré-Condição: O utilizador identifica-se e o sistema percebe que este tem os devidos privilégios associados e por isso apresenta-lhe o formulário para abrir manualmente uma caixa e definir o fundo de maneio.
<b>Saída e pós-condições:</b>	Saída: ---
	Pós-Condições: A partir deste momento todas sessões abertas e associadas a esta caixa irão realizar movimentos sobre o fundo de maneio disponibilizado aqui.
<b>Diagrama Caso de Uso</b>	<p>The diagram shows an actor labeled 'Employee with privileges' connected by a line to a use case labeled 'Open CashBox'.</p>

[RF011] FECHAR CAIXA

<b>Este caso de uso permite que um utilizador com privilégios feche manualmente uma caixa previamente aberta. Por fechar caixa entenda-se o fechar da sessão da caixa actualmente em curso.</b>	
<b>Prioridade:</b>	Essencial
<b>Entrada e pré-condições:</b>	Entrada: Instante temporal em que é fechada a caixa e a referência para a caixa em questão.
	Pré-Condição: O utilizador identifica-se e o sistema percebe que este tem os devidos privilégios associados e por isso apresenta-lhe o formulário para fechar manualmente uma caixa lógica.
<b>Saída e pós-condições:</b>	Saída: ---
	Pós-Condições: A partir deste momento todas sessões abertas e associadas a esta caixa irão ser fechadas automaticamente, a não ser que existam vendas activas. Nesse caso deverá ser perguntado ao utilizador como é que ele quer tratar as vendas activas.
<b>Diagrama Caso de Uso</b>	<p>The diagram shows an actor labeled 'Employee with privileges' connected by a line to a use case labeled 'Close Cashbox'.</p>

[RF012] INICIAR SESSÃO DE EMPREGADO

<b>Este caso de uso permite que um funcionário inicie a sua sessão no sistema.</b>	
<b>Prioridade:</b>	Essencial
<b>Entrada e pré-condições:</b>	Entrada: Instante temporal em que é iniciada a sessão e os dados do funcionário contemplado.
	Pré-Condição: O utilizador autentica-se com os seus dados no sistema e a sua sessão é iniciado.
<b>Saída e pós-condições:</b>	Saída: ---
	Pós-Condições: A partir deste momento o empregado passa a poder realizar operações no sistema e todos os movimentos realizados ficarão associados à sessão acabada de criar.



[RF013] FECHAR SESSÃO DE EMPREGADO

<b>Este caso de uso permite que um funcionário finalize manualmente a sua sessão.</b>	
<b>Prioridade:</b>	Essencial
<b>Entrada e pré-condições:</b>	Entrada: Instante temporal em que é fechada a sessão e a sessão contemplada.
	Pré-Condição: O utilizador preenche o formulário para fechar manualmente a sua sessão.
<b>Saída e pós-condições:</b>	Saída: ---
	Pós-Condições: A partir deste momento o empregado deixa de poder realizar qualquer operação no sistema.
<b>Diagrama Caso de Uso</b>	

[RF014] TRANSFERÊNCIA DE VENDAS ACTIVAS

<b>Este caso de uso permite que um utilizador com privilégios transfira as vendas activas da sessão de um empregado para a sessão de outro empregado.</b>	
<b>Prioridade:</b>	Importante
<b>Entrada e pré-condições:</b>	Entrada: Sessão do empregado de origem e a sessão do empregado de destino.
	Pré-Condição: O utilizador identifica-se e o sistema percebe que este tem os devidos privilégios associados e por isso apresenta-lhe o formulário para realizar a transferência de vendas de uma sessão para outra.
<b>Saída e pós-condições:</b>	Saída: ---
	Pós-Condições: A partir deste momento a sessão do empregado de destino irá incluir as vendas activas que constavam da sessão do empregado de origem. Para além disso a sessão do empregado de origem será fechada automaticamente.
<b>Diagrama Caso de Uso</b>	

... (Página 91/94) ...

## Matriz de requisitos funcionais por prioridade

Código Requisito	Essencial	Importante	Desejável
RF001 - Criar família ou subfamília de produtos	<input checked="" type="checkbox"/>		
RF002 - Procurar família ou subfamília de produtos	<input checked="" type="checkbox"/>		
RF003 - Editar família ou subfamília de produtos	<input checked="" type="checkbox"/>		
RF004 - Eliminar família ou subfamília de produtos	<input checked="" type="checkbox"/>		
RF005 - Criar artigo	<input checked="" type="checkbox"/>		
RF006 - Procurar artigo	<input checked="" type="checkbox"/>		
RF007 - Editar artigo	<input checked="" type="checkbox"/>		
RF008 - Eliminar artigo	<input checked="" type="checkbox"/>		
RF009 - Criar Complemento	X	<input checked="" type="checkbox"/>	
RF010 - Procurar Complemento	X	<input checked="" type="checkbox"/>	
RF011 - Editar Complemento	X	<input checked="" type="checkbox"/>	
RF012 - Eliminar Complemento	X	<input checked="" type="checkbox"/>	
RF013 - Criar Serviço			<input checked="" type="checkbox"/>
RF014 - Editar Serviço			<input checked="" type="checkbox"/>
RF015 - Procurar Serviço			<input checked="" type="checkbox"/>
RF016 - Eliminar Serviço			<input checked="" type="checkbox"/>
RF017 - Criar Atributo/Valor do Item de Venda			<input checked="" type="checkbox"/>
RF018 - Editar Atributo/Valor do Item de Venda			<input checked="" type="checkbox"/>
RF019 - Remover Atributo/Valor do Item de Venda			<input checked="" type="checkbox"/>
RF020 - Unidades Predefinidas	<input checked="" type="checkbox"/>		
RF021 - Criar Unidade			<input checked="" type="checkbox"/>
RF022 - Procurar Unidade			<input checked="" type="checkbox"/>
RF023 - Editar Unidade			<input checked="" type="checkbox"/>
RF024 - Eliminar Unidade			<input checked="" type="checkbox"/>
RF025 - Criar Taxa	<input checked="" type="checkbox"/>		
RF026 - Procurar Taxa	<input checked="" type="checkbox"/>		
RF027 - Editar Taxa	<input checked="" type="checkbox"/>		
RF028 - Eliminar Taxa	<input checked="" type="checkbox"/>		
RF029 - Definir Preferências Regionais	<input checked="" type="checkbox"/>		
RF030 - Definir exclusões	X		<input checked="" type="checkbox"/>
RF031 - Criar Identificador	<input checked="" type="checkbox"/>		
RF032 - Procurar Identificador	<input checked="" type="checkbox"/>		
RF033 - Editar Identificador	<input checked="" type="checkbox"/>		
RF034 - Eliminar Identificador	<input checked="" type="checkbox"/>		
RF035 - Editar Serie de Documentos Manualmente	<input checked="" type="checkbox"/>		
RF036 - Editar Serie de Documentos Automaticamente			<input checked="" type="checkbox"/>
RF037 - Imprimir Documento	<input checked="" type="checkbox"/>		
RF038 - Procurar Documentos	<input checked="" type="checkbox"/>		
RF039 - Tipos de documentos predefinidos	<input checked="" type="checkbox"/>		
RF040 - Definir método de pag. Do tipo de documento	<input checked="" type="checkbox"/>		
RF041 - Layouts Pré-definidos por tipos de documentos	<input checked="" type="checkbox"/>		
RF042 - Associar Layout ao tipo de documento			<input checked="" type="checkbox"/>
RF043 - Criar Atributo/Valor do documento			<input checked="" type="checkbox"/>
RF044 - Editar Atributo/Valor do documento			<input checked="" type="checkbox"/>
RF045 - Remover Atributo/Valor do documento			<input checked="" type="checkbox"/>
RF046 - Criar Utilizador	<input checked="" type="checkbox"/>		
RF047 - Procurar Utilizador	<input checked="" type="checkbox"/>		
RF048 - Eliminar Utilizador	<input checked="" type="checkbox"/>		
RF049 - Editar Utilizador	<input checked="" type="checkbox"/>		
RF050 - Definir Grupo do Utilizador	<input checked="" type="checkbox"/>		
RF051 - Gestão de Férias/Folgas do Funcionário			<input checked="" type="checkbox"/>
RF052 - Alterar língua para o utilizador	X		<input checked="" type="checkbox"/>
RF053 - Criar Atributo/Valor do utilizador			<input checked="" type="checkbox"/>
RF054 - Editar Atributo/Valor do utilizador			<input checked="" type="checkbox"/>

RF055 – Remover Atributo/Valor do utilizador			<input checked="" type="checkbox"/>
RF056 - Criar Grupo de Utilizadores	<input checked="" type="checkbox"/>		
RF057 - Editar Grupo de Utilizadores	<input checked="" type="checkbox"/>		
RF058 - Procurar Grupo de Utilizadores	<input checked="" type="checkbox"/>		
RF059 - Remover Grupo de Utilizadores	<input checked="" type="checkbox"/>		
RF060 - Definir Privilégios de um Grupo de Utilizadores		<input checked="" type="checkbox"/>	
RF061 – Criar Promoção			<input checked="" type="checkbox"/>
RF062 - Editar Promoção			<input checked="" type="checkbox"/>
RF063 - Procurar Promoção			<input checked="" type="checkbox"/>
RF064 - Remover Promoção			<input checked="" type="checkbox"/>
RF065 – Criar Menu			<input checked="" type="checkbox"/>
RF066 – Editar Menu			<input checked="" type="checkbox"/>
RF067 - Procurar Menu			<input checked="" type="checkbox"/>
RF068 – Remover Menu			<input checked="" type="checkbox"/>
RF069 – Criar Campanha de Desconto			<input checked="" type="checkbox"/>
RF070 - Editar Campanha de Desconto			<input checked="" type="checkbox"/>
RF071 - Procurar Campanha de Desconto			<input checked="" type="checkbox"/>
RF072 - Remover Campanha de Desconto			<input checked="" type="checkbox"/>
RF073 – Criar Local	<input checked="" type="checkbox"/>		
RF074 - Editar Local	<input checked="" type="checkbox"/>		
RF075 - Procurar Local	<input checked="" type="checkbox"/>		
RF076 - Remover Local	<input checked="" type="checkbox"/>		
RF077 - Definir Ecrã de Selecção Rápida			<input checked="" type="checkbox"/>
RF078 - Criar Caixa/Gaveta	<input checked="" type="checkbox"/>		
RF079 - Editar Caixa/Gaveta	<input checked="" type="checkbox"/>		
RF080 - Procurar Caixa/Gaveta	<input checked="" type="checkbox"/>		
RF081 - Remover Caixa/Gaveta	<input checked="" type="checkbox"/>		
RF082 - Definir valor do Envelope da Caixa	<input checked="" type="checkbox"/>		
RF083 – Realizar operação de entrada em caixa	<input checked="" type="checkbox"/>		
RF084 – Realizar operação de saída de caixa	<input checked="" type="checkbox"/>		
RF085 - Registar Impressora	<input checked="" type="checkbox"/>		
RF086 - Editar Impressora	<input checked="" type="checkbox"/>		
RF087 - Procurar Impressora	<input checked="" type="checkbox"/>		
RF088 - Remover Impressora	<input checked="" type="checkbox"/>		
RF089 – Testar Impressora	<input checked="" type="checkbox"/>		
RF090 - Adicionar Terminal	<input checked="" type="checkbox"/>		
RF091 - Editar Terminal			<input checked="" type="checkbox"/>
RF092 - Procurar Terminal			<input checked="" type="checkbox"/>
RF093 – Criar Atributo/Valor do dispositivo			<input checked="" type="checkbox"/>
RF094 – Editar Atributo/Valor do dispositivo			<input checked="" type="checkbox"/>
RF095 – Remover Atributo/Valor do dispositivo			<input checked="" type="checkbox"/>
RF096 - Abrir venda activa	<input checked="" type="checkbox"/>		
RF097 - Registar Pedido	<input checked="" type="checkbox"/>		
RF098 - Anular Pedido	<input checked="" type="checkbox"/>		
RF099 - Registar Pagamento	<input checked="" type="checkbox"/>		
RF100 - Realizar Estorno	<input checked="" type="checkbox"/>		
RF101 - Realizar Reembolso	<input checked="" type="checkbox"/>		
RF102 - Fechar turno			<input checked="" type="checkbox"/>
RF103 - Iniciar sessão da loja (Dia)	<input checked="" type="checkbox"/>		
RF104 - Fechar sessão da loja (Dia)	<input checked="" type="checkbox"/>		
RF105 - Abrir Caixa	<input checked="" type="checkbox"/>		
RF106 - Fechar Caixa	<input checked="" type="checkbox"/>		
RF107 - Iniciar sessão de Empregado	<input checked="" type="checkbox"/>		
RF108 - Fechar sessão de Empregado	<input checked="" type="checkbox"/>		
RF109 - Transferência de vendas activas		<input checked="" type="checkbox"/>	
RF110 - Criar Mesa			<input checked="" type="checkbox"/>
RF111 - Editar Mesa			<input checked="" type="checkbox"/>
RF112 - Procurar Mesa			<input checked="" type="checkbox"/>
RF113 - Remover Mesa			<input checked="" type="checkbox"/>



RF114 - Criar Cartão	<input checked="" type="checkbox"/>		
RF115 - Editar Cartão	<input checked="" type="checkbox"/>		
RF116 - Procurar Cartão	<input checked="" type="checkbox"/>		
RF117 - Remover Cartão	<input checked="" type="checkbox"/>		
RF118 - Criar Reserva			<input checked="" type="checkbox"/>
RF119 - Editar Reserva			<input checked="" type="checkbox"/>
RF120 - Procurar Reserva			<input checked="" type="checkbox"/>
RF121 - Remover Reserva			<input checked="" type="checkbox"/>
RF122 - Gerar Relatório de Gestão de apoio ao negócio		<input checked="" type="checkbox"/>	
RF123 - Eliminar Relatório de Gestão de apoio ao negócio		<input checked="" type="checkbox"/>	
RF124 - Procurar Relatório		<input checked="" type="checkbox"/>	
RF125 - Imprimir Relatório		<input checked="" type="checkbox"/>	
RF126 – Criar Evento			<input checked="" type="checkbox"/>
RF127 – Editar Evento			<input checked="" type="checkbox"/>
RF128 – Procurar Evento			<input checked="" type="checkbox"/>
RF129 – Remover Evento			<input checked="" type="checkbox"/>
RF130 – Consultar Notificações			<input checked="" type="checkbox"/>
RF131 – Criar Grupo de Comissão			<input checked="" type="checkbox"/>
RF132 – Editar Grupo de Comissão			<input checked="" type="checkbox"/>
RF133 – Procurar Grupo de Comissão			<input checked="" type="checkbox"/>
RF134 – Remover Grupo de Comissão			<input checked="" type="checkbox"/>
RF135 – Criar Regra de Comissão			<input checked="" type="checkbox"/>
RF136 – Editar Regra de Comissão			<input checked="" type="checkbox"/>
RF137 – Procurar Regra de Comissão			<input checked="" type="checkbox"/>
RF138 – Remover Regra de Comissão			<input checked="" type="checkbox"/>
RF139 – Criar Cliente			<input checked="" type="checkbox"/>
RF140 – Editar Cliente			<input checked="" type="checkbox"/>
RF141 – Procurar Cliente			<input checked="" type="checkbox"/>
RF142 – Remover Cliente			<input checked="" type="checkbox"/>
RF143 – Criar Atributo/Valor do Cliente			<input checked="" type="checkbox"/>
RF144 – Editar Atributo/Valor do Cliente			<input checked="" type="checkbox"/>
RF145 – Remover Atributo/Valor do Cliente			<input checked="" type="checkbox"/>
RF146 – Criar Mensagem Template			<input checked="" type="checkbox"/>
RF147 – Editar Mensagem Template			<input checked="" type="checkbox"/>
RF148 – Remover Mensagem Template			<input checked="" type="checkbox"/>
RF149 – Enviar Mensagem			<input checked="" type="checkbox"/>
RF150 – Consultar Mensagem			<input checked="" type="checkbox"/>
RF151 – Perfis de Clientes Pré-definidos	<input checked="" type="checkbox"/>		
RF152 – Criar Perfil de Cliente			<input checked="" type="checkbox"/>
RF153 - Editar Perfil de Cliente			<input checked="" type="checkbox"/>
RF154 - Remover Perfil de Cliente			<input checked="" type="checkbox"/>
RF155 – Criar Zona de Impressão			<input checked="" type="checkbox"/>
RF156 – Editar Zona de Impressão			<input checked="" type="checkbox"/>
RF157 – Remover Zona de Impressão			<input checked="" type="checkbox"/>
RF158 – Editar Loja			<input checked="" type="checkbox"/>

→ Prioridade definida de acordo com *Softkom*.

X → Prioridade revista pela equipa de desenvolvimento.

## ANEXO B

Excerto do documento com especificação funcional inicial, produzido no âmbito do desenvolvimento do projecto SIGER.

... (Página 35/97) ...

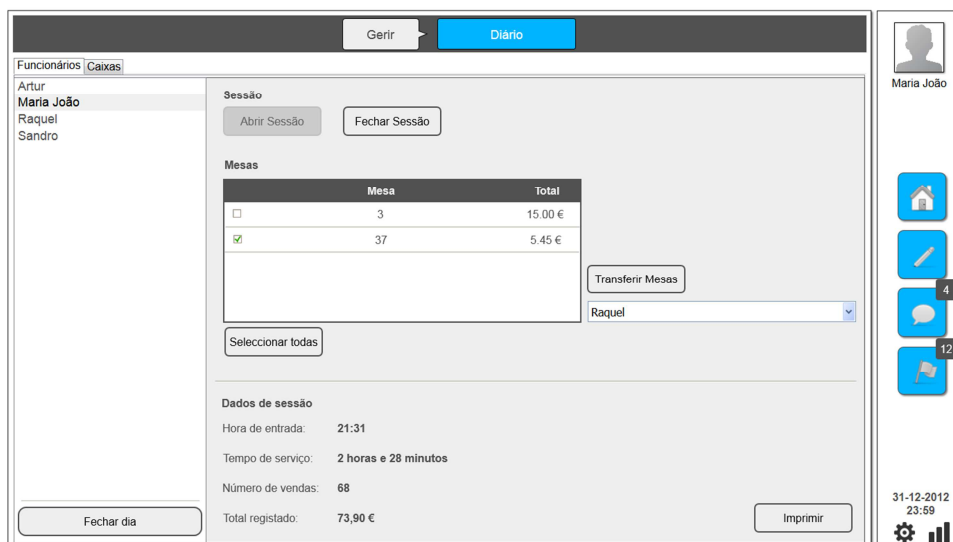
### Gerir Diário

#### Funcionalidades

- Gerir sessões
  - Sessão de funcionário
    - Abrir sessão de funcionário
    - Fechar sessão de funcionário
    - Transferir vendas activas
  - Caixa
    - Abrir caixa
    - Fechar caixa
    - Configurar abertura automática de caixa
  - Transferência de Turno
  - Fechar dia

Nesta área o utilizador pode gerir o estado das caixas e dos funcionários no estabelecimento.

#### Sessões de funcionário

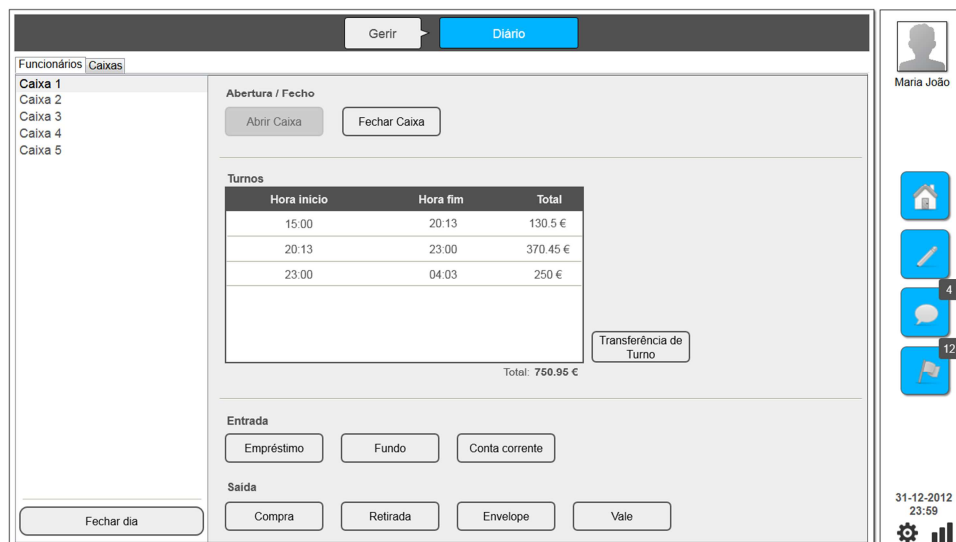


Mockup 1 - Gestão de funcionários

Cada funcionário tem obrigatoriamente de abrir sessão antes de efectuar qualquer operação no sistema. No final do seu horário de trabalho o funcionário deve fechar a sua sessão e transferir as mesas que lhes estão associadas para outro funcionário. Para isso basta seleccionar o funcionário de destino e carregar em *transferir mesas*. De notar que a transferência de mesas só está disponível de acordo com as permissões do utilizador que vai transferir as mesas e das permissões do funcionário para o qual vão ser transferidas.

No fundo da área de gestão de sessão de funcionários são apresentados dados da sua sessão, tais como: hora de entrada, tempo de serviço, o número de vendas registadas e o total que essas vendas perfazem.

## Caixas



Mockup 2 - Gestão de caixa

### Abertura e fecho de caixa

Antes de poder efectuar qualquer operação numa caixa é necessário abri-la no sistema. Para isso basta seleccionar a caixa e carregar em *Abrir caixa*. As caixas podem também ser abertas automaticamente com a abertura do dia, sendo essa configuração efectuada na área de gestão de caixas. No final do dia, o funcionário responsável deverá fechar a caixas, seleccionando a opção *Fechar caixa*.

### Turnos

A qualquer altura do dia pode ser feita uma transferência de turno de uma caixa. Para isso o utilizador deve seleccionar a caixa e carregar em *Transferência de turno*. De seguida é apresentada uma pequena janela para o utilizador introduzir o valor real que contou em caixa, e é dada a informação se esse valor corresponde ao que está registado no sistema para aquela caixa.

### Entrada

Nesta secção são dadas as opções possíveis para adicionar dinheiro numa das caixas. No caso do utilizador estar a adicionar dinheiro dele na caixa deve carregar em *Empréstimo* e indicar o valor que está a colocar. Se o dinheiro que está a adicionar é do estabelecimento deve escolher a opção *Fundo* e indicar o valor que está a inserir. Se o dinheiro que está a adicionar à caixa é referente ao pagamento de uma conta corrente o utilizador deve carregar no botão correspondente, procurar a conta em causa e registar o pagamento.

### Saída

Se for necessário retirar algum dinheiro da caixa, o utilizador deve escolher a razão correspondente:

- Compra

- Retirada
- Envelope
  - Se o valor em caixa atingir o máximo estabelecido o utilizador deverá remover uma quantia e registar o montante no sistema através desta opção
- Vale
  - Valor emprestado a um dos funcionários. O valor do vale deve ser registado através desta opção

A distinção entre os vários tipos de saída de dinheiro é importante pois tem implicações em termos fiscais.

### Fechar dia

Depois de fechar todas as sessões de funcionário e caixas é possível fechar o dia. Nesse momento é registado e emitido um documento com os dados do dia.

De notar que cada utilizador terá um conjunto de permissões que permitirá ou não executar as operações acima descritas. Um administrador terá de ter hipótese de fechar o dia e transferir turnos, no entanto isso poderá não se aplicar aos funcionários e essas opções nem lhes serão apresentadas.

- Gerir Famílias
- Funcionalidades
- Gerir famílias
- Adicionar Família
- Editar família
- Eliminar família

Este ecrã permite ao utilizador gerir as famílias existentes no sistema. Na esquerda é apresentada a árvore de famílias existentes, podendo esta ser editada de acordo com as necessidades do negócio.

Mockup 3 - Gestão de famílias

## Adicionar

Ao carregar no botão *Novo*, é criada uma nova família filha da família seleccionada. Se nenhuma estiver seleccionada é criada uma família *raiz*, podendo este campo ser alterado no painel principal.

## Editar

O utilizador pode tocar na família que deseja editar e posteriormente fazer as alterações que desejar no painel principal. Depois de editar a família o utilizador deve carregar em *Guardar* para que as alterações tomem efeito.

## Eliminar

Da mesma forma que o utilizador pode editar, pode também eliminar uma família seleccionando-a na lista e carregando no botão *Eliminar*. Importante notar que as famílias só podem ser editadas ou eliminadas se não estiverem a ser utilizadas por nenhum Item de venda, pois caso exista alguma dependência o utilizador não poderá efectuar a operação.

É possível definir que famílias estão disponíveis em que locais e para que funcionários, no entanto essa gestão é feita na área de gestão de exclusões que é descrita no capítulo “Gerir Exclusões”.

- Gerir Itens de Venda
- Funcionalidades
- Gerir itens de venda
- Adicionar item de venda
- Editar item de venda
- Eliminar item de venda

Neste ecrã o utilizador pode adicionar, editar ou eliminar um item de venda. Na barra da esquerda são listados por categoria os itens já existentes no sistema.

## Definições gerais

The screenshot displays the 'Gerir Itens de Venda' (Manage Sales Items) interface. On the left, a sidebar lists categories: Artigos (Cafeteria, Café, Descafeinado, Água, Água das pedra, Água castelo, Freeze, Pastel de nata, Carne, Peixe), Complementos, and Serviços. The main area shows the configuration for a 'Descafeinado' item. Fields include: Nome (Descafeinado), Descrição, Código (DESC), Cód. Via Rádio (2), Família (Cafeteria), Zona de impressão (Zona 1), Preço (omissão) (0.60 €), Preço Compra (0.10 €), Margem de lucro (30%), and Taxas (IVA AGRAVADO (23%), Taxa 1). A photo of a coffee cup is shown. The bottom navigation bar includes 'Geral', 'Frações', 'Preço', and 'Complementos'. The right sidebar shows the user profile 'Maria João', navigation icons, and the date/time '31-12-2012 23:59'.

Mockup 4 - Gestão de itens de venda

Para expandir uma das categorias (artigos, complementos ou serviços) o utilizador deve tocar nela, sendo assim apresentado o seu conteúdo. Para adicionar um item de venda o utilizador deve carregar no botão *Adicionar*. Será criado um novo item de venda dentro da categoria seleccionada, devendo preencher os seguintes campos de acordo com o desejado.

- Nome
- Descrição
- Código
  - Consiste numa abreviatura ou código rápido do item que será útil em pesquisas ou na identificação de artigos em espaços reduzidos onde não é possível mostrar o nome completo do artigo
- Código Via Rádio
  - Código numérico a ser utilizado pelos comandos via rádio
- Família
- Zona de impressão
  - Local onde será impresso/mostrado o pedido se necessário
- Se o item é vendido à fracção
  - Indica se aquele item pode ser quantidades diferentes com preços individuais, ou seja, por exemplo a cerveja Sagres pode ser vendida em fracções de 25cl ou 33cl, cada uma com um preço diferente.
- Preço por omissão
  - Preço a ser utilizado quando mais nenhum está definido
- Preço de compra
- Margem de lucro
- Se o preço é variável
  - Indica se o preço pode ser alterado no momento da venda
- Se o item é vendido à fracção
  - No caso do item ser vendido à fracção o utilizador deve carregar na aba *Fracção* e definir as propriedades de cada fracção

### **Taxas**

Percentagem a ser aplicada sobre o preço estipulado. Esta taxa é aplicada independentemente de qualquer alteração que possa ser feita no preço através de uma promoção.

### **Foto**

Para escolher a foto do artigo o utilizador deve carregar sobre a área da foto e escolher o ficheiro desejado

### **Fracções**

Caso o item seja vendido a fracção, o utilizador deve seleccionar qual a unidade base das fracções, as fracções propriamente ditas e o seu respectivo preço.

Gerir Itens de Venda

Unidade base: centilitro - cl

Fracção variável (deve ser definida fracção base)

Fracção	Preço
20 cl	0.70 €
40 cl	1.5 €
50 cl	3 €

Artigos:

- ▼ Cafeteria
  - Café
  - Descafeinado
  - Água
  - Água das pedra
  - Água castelo
- ▶ Bebidas
  - Cerveja pressão**
  - Carne
  - Peixe
- Complementos
- Serviços

Adicionar Eliminar Guardar

31-12-2012 23:59

Mockup 5 - Gerir fracções de item de venda

Se o utilizador indicar que as fracções são variáveis, o sistema irá obrigar a criar apenas uma fracção com o respectivo preço, que servirá de base para o cálculo dos preços das fracções registadas no momento da venda. Esta configuração é útil para item de sejam vendidos ao peso, tais como peixe, carne, etc.

## Preços

O utilizador pode definir vários preços para o mesmo item no separador preço, devendo para isso criar um novo preço para a combinação de 2 campos: local e data/hora

Gerir Itens de Venda

Quantidade	Preço
20 cl	0.70 €
40 cl	1.5 €
50 cl	3 €

Preço	Configuração
0.80 €	ESP

Local:

- Todos [TDS]
- Balcão 1 [B1]
- Balcão 2 [B2]
- Balcão 3 [B3]
- Esplanada [ESP]

Início:

Data: DD-MM-AAAA

Hora: -- --

Novo Preço: 0.80 €

Adicionar Eliminar Guardar

31-12-2012 23:59

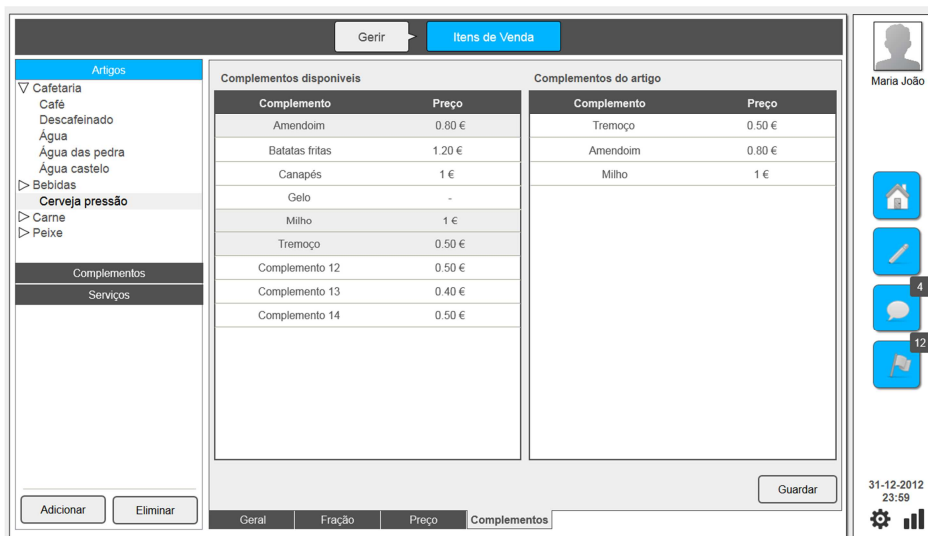
Mockup 6 - Gestão de preço de item de venda

Se por exemplo for definido o preço de 0.80€ para o artigo “Cerveja pressão”, no local “Esplanada” e não for definida nenhuma data, qualquer pedido desse artigo na esplanada irá ser cobrado ao preço de 0.80€. Para guardar o item o utilizador deverá carregar no botão *Guardar*, só assim as alterações tomaram efeito no sistema.

Se criar um novo item numa família onde já existirem outros itens, as suas características genéricas são copiadas. Isto significa que o utilizador só terá de definir manualmente as características (taxas, unidade base e estado) no primeiro item da família, os seguintes herdam estas características, podendo se necessário serem alteradas.

## Complementos

A aba complementos só é apresentada se o item de venda que estiver a ser editado pertencer a categoria de artigos. Este tipo de item de venda pode ter complementos associados. Neste ecrã são listados todos os complementos existentes no sistema e respectivo preço, podendo associa-los ao artigo seleccionado.



Mockup 7 - Gestão de complementos

Para associar um complemento a um artigo basta que o utilizador carregue sobre o complemento desejado, este passará para a lista da direita e sombreado na lista da esquerda, indicando desta forma que já está a ser utilizado. Para remover a associação, o utilizador deve carregar sobre o complemento na lista da direita, este desaparecerá e ficará novamente disponível na lista da esquerda.

Ao tocar num dos itens de venda são apresentadas no painel central as suas características, podendo estas ser editadas livremente pelo utilizador. As alterações só tomaram efeito depois de carregar no botão *Guardar*.

Para eliminar um artigo o utilizador deve selecciona-lo e de seguida carregar em *Eliminar*. Nesse momento será feita a validação se não existe nenhuma promoção dependente do item em causa, se tal se verificar o item é eliminado, caso contrário o utilizador é alertado para esse facto.

... (Página 63/97) ...

## Sistema de Controlo de DD

O sistema de controlo de DD tem como objectivo permitir numa fase inicial do projecto, provar e testar as funcionalidades chave do produto final:

- Estado do sistema
- Analisar consumos por DD
- Analisar consumos por bebidas
- Histórico de consumos/vendas (consumos por hora do dia, por DD, por tipo bebida)
- Gestão de alertas e relatórios periódicos



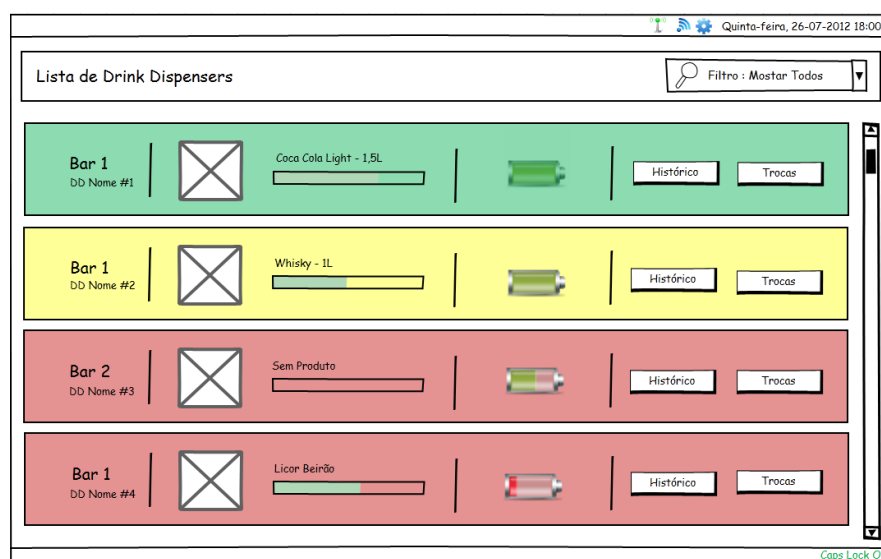
- Ver número de mudanças de garrafa por dispositivo
- Estado de cada DD: bateria, presença garrafa, doses servidas
- Troca de garrafa de um DD
- Criar níveis de alerta para falta de bateria
- Registrar consumos

Em suma, dar ao proprietário do estabelecimento uma visão rigorosa e um controlo profundo do estado do seu negócio, de forma a maximizar os seus lucros e minimizar prejuízos. Para isso, a interacção entre o POS e os funcionários do estabelecimento deve ser melhorada e toda a informação deve chegar à administração através de uma *interface* simplista e *user-friendly*, mas ao mesmo tempo completa.


Nos capítulos seguintes são apresentados os *mockups* do sistema de gestão de DD, dividindo-os entre o que é destinado à administração e aos funcionários do estabelecimento. Existe no entanto uma zona comum ao administrador e ao funcionário que consiste na monitorização do estado do sistema, onde rápida e facilmente se pode concluir onde está a causa de um possível problema.


## Estado do Sistema


As barras de topo e de fundo são áreas comuns a toda a aplicação onde é mostrada informação acerca do estado actual do sistema.



Mockup 1 - Estado do Sistema

No canto superior direito é mostrado o dia da semana, o dia do mês e a hora actual, indicando cada um dos icons à esquerda um estado específico de um módulo do sistema. O icon  indica o estado do ciclo de CPU, e desde que se encontre a rodar a máquina estará a funcionar correctamente.

O icon  indica o estado da rede, que quanto mais traços azuis forem visíveis melhor será o sinal, sendo que no caso de não ter qualquer traço existe um problema de rede.

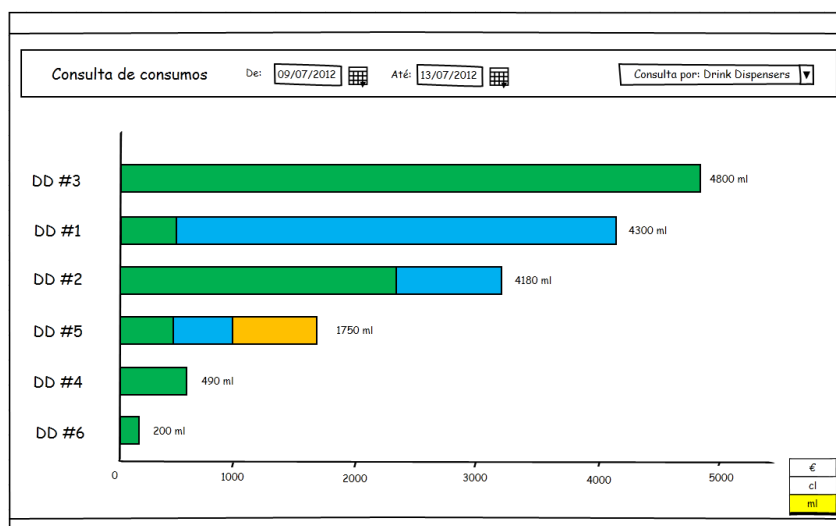
O icon  indica o estado de comunicação com o *Master Receiver*. Se o icon for apresentado a verde significa que o sistema está a comunicar correctamente com o dispositivo, se este se encontrar a vermelho significa que existe um problema de comunicação que deve ser resolvido.

Por último, no canto inferior esquerdo é dada a indicação se o Caps Lock está ligado ou não. Case se encontre ligado é apresentado o texto “Caps Lock Ligado” a cor verde, caso contrário a mensagem apresentada é “Caps Lock Desligado” e é escrita a cor cinza.

## Área de Administração

O sistema de administrador permite ter uma noção do estado do negócio da perspectiva do proprietário, permitindo-lhe tomar opções de acordo com os interesses do seu estabelecimento. Neste sistema pode consultar os **consumos em função dos DD e das bebidas**, dando-lhe uma visão geral do estado dos consumos. Para ter uma visão mais detalhada das vendas, existem também **gráficos dos consumos daquele DD ou daquela bebida num determinado dia**, podendo ver no mesmo gráfico os consumos de DD/bebidas diferentes por forma a comparar os resultados. Neste sistema pode ainda ser consultado o **estado detalhado de cada DD** (número de mudanças de garrafa, bateria, doses servidas).

### Analisar consumos por DD

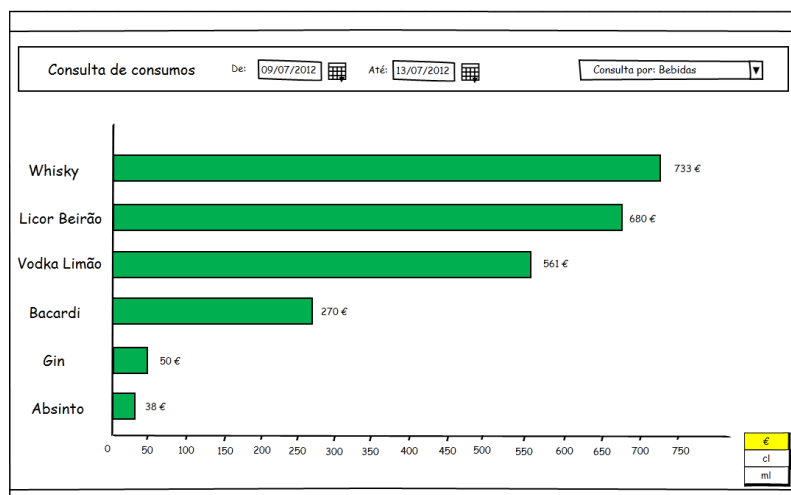


Mockup 2 - Consumos em ml de cada DD registados no sistema

Os consumos em função dos DD são apresentados por ordem decrescente, sendo que cada barra horizontal corresponde ao consumo efectuado por aquele DD no intervalo de tempo seleccionado. Como os DD podem ser movidos entre garrafas de produtos diferentes, podem existir cores diferentes na mesma barra indicando que isso aconteceu.

Esta área permite fazer a gestão de consumos em função dos cl/ml servidos ou em função do lucro obtido a partir desses serviços, bastando para alternar entre estas unidades seleccionar a unidade pretendida na *combo box* no canto inferior direito.

### Analisar consumos por bebidas



Mockup 3 - Consumos em Euros de cada produto registado no sistema

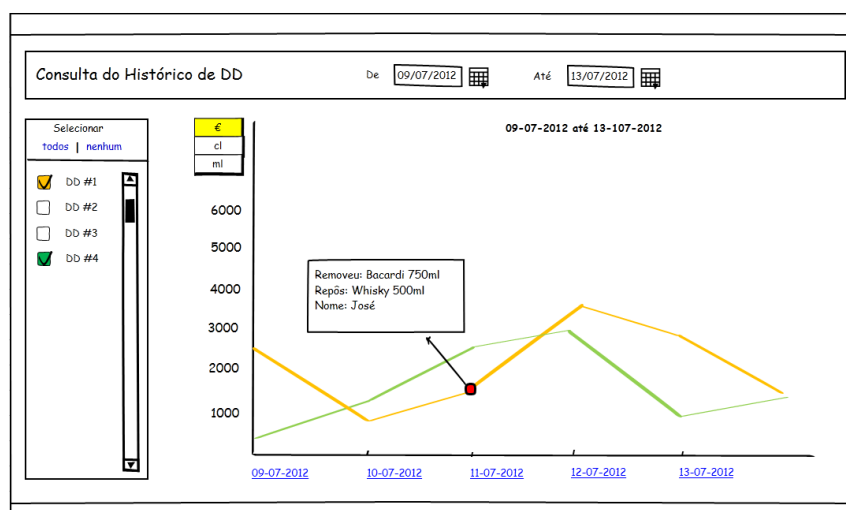
Semelhante à consulta de consumos por DD, sendo que nesta área da aplicação os consumos são mostrados em função das bebidas servidas.

Tal como na gestão de consumos por DD é possível ver os consumos num total de cl/ml servidos ou no total de facturação entre as datas que foram seleccionadas.

### Histórico de consumos/vendas

Ao contrário da visão macro da consulta de consumos anteriores, nesta área é dada uma visão detalhada dos consumos de um ou mais DD/bebidas num intervalo de tempo. O detalhe dessa amostragem é por defeito ao nível do dia, podendo ser alterado para o nível da hora se o utilizador clicar num dos dias. No caso do histórico ser consultado a partir da gestão de consumos (Figura 2 e 3), o intervalo de tempo é mantido de uma janela para a outra. No caso do histórico ser directamente acedido, o intervalo de tempo é definido para os últimos 7 dias. O utilizador pode também alternar entre consultar os consumos por cl/ml ou por Euros.

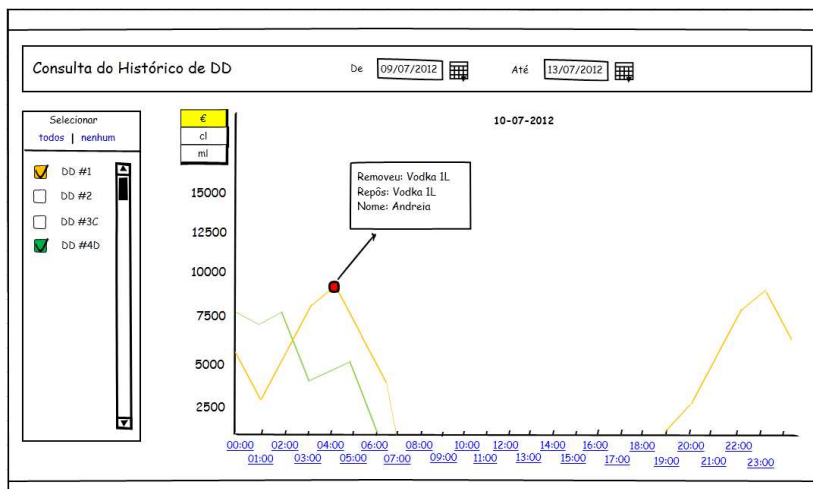
- Histórico por DD



Mockup 4 - Histórico de consumos de DD por dia

Neste gráfico são mostrados os consumos em ml dos DD 1 e 4, para os dias entre 9-07-2012 e 13-07-2012. Caso o utilizador pretenda ver os consumos de mais DD simultaneamente apenas é necessário seleccionar *checkbox* pretendida. As trocas de garrafa que possam existir são

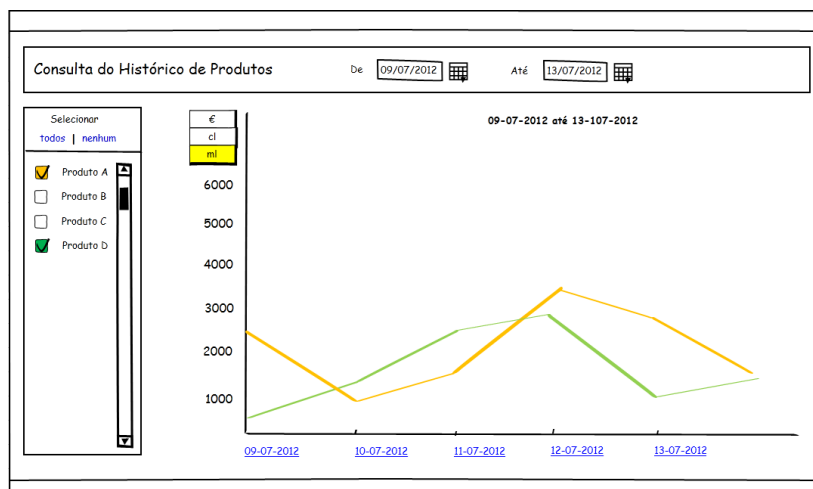
assinaladas por pontos ao longo do gráfico, que ao carregar com o dedo/rato mostra o produto retirado, o produto reposto e quem foi o funcionário que o fez.



Mockup 5 - Histórico de DD num dia específico

Para ver os consumos de um dia específico, o dia pretendido no fundo do gráfico deve ser clicado. No gráfico por dia também é possível ver as trocas de garrafa que foram feitas num determinado DD.

- Histórico por produtos



Mockup 6 - Histórico de consumo de produtos por dia

Semelhante ao histórico por DD, mostra os consumos em cl/ml dos produtos A e B, entre os dias 9-07-2012 e 13-07-2012. Podem também ser adicionados no mesmo gráfico consumos de produtos diferentes, bastando para isso seleccionar as *checkbox* na lista de produtos. Ao clicar num dos dias no fundo do gráfico, é feito o *drill down* para esse dia apresentado os consumos por hora efectuados nesse dia específico.