**Relatório Final**

**C:\Users\Eduardo\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\27.tif**

**Conferência e Recepção de Material**

Sistemas de Informação

MIEIC

2014

**Grupo :** João Pascoal Faria 201106849 ei11103@fe.up.pt

Luís Abreu 201106789 ei11146@fe.up.pt

Pavel Alexeenko 201104331 ei11155@fe.up.pt

Pedro Silva 201109244 ei11061@fe.up.pt

Rui Grandão Rocha 201004302 ei11010@fe.up.pt

17 de dezembro de 2014

**Índice**

1. Descrição do Projeto
2. *User Stories*
3. *Solution Framework* e Tecnologias Adotadas
4. *Core Views*
5. Arquitetura Tecnológica e Funcional
6. Instruções de Utilização
   1. Autenticação na Aplicação
   2. Execução da Máquina Virtual
7. Referências

**Cenas escritas no enunciado do trabalho**

**Descrição Macro:** Criar uma aplicação html5 que permita fazer a conferência de recepção de material e os respetivos acertos, caso existam diferenças.

**Objetivos de extensibilidade do ERP:** API para autenticação, consulta de encomenda, satisfação de encomenda, entrada de armazém, gerar doc. Remessa fornecedor.

**Objetivos Utilização do ERP:** 1. Correta definição de dados mestre no ERP (fornecedores, armazéns, artigos, categorias) /// 2. Criação de encomendas /// 3. No ERP deverão concluir o processo com uma factura de fornecedor e pagamento

**1. Descrição do Projeto**

O projeto desenvolvido pretende ser uma extensão do Sistema de Informação desenvolvido pela empresa Primavera: o Primavera Profissional ERP.

Um sistema de informação é um conjunto de componentes integrados com o objetivo de armazenar e processar dados para, posteriormente, devolver informações e análises relevantes para um negócio. Muitas empresas e organizações utilizam sistemas como este para realizar e gerir as suas operações, interagir com os seus clientes e fornecedores, ao mesmo tempo que se mantém competitivos no mercado[1].

Para o controlo de gestão comercial de matérias primas e/ou produtos finais de uma empresa são necessários espaços de armazenamento sobre os quais, os funcionários responsáveis pela recepção de encomendas têm de efetuar a verificação e validação dos produtos recepcionados. Porém, essa validação pode tornar-se longa e demorada devido à complexidade dos sistemas de informação, podendo gerar facilmente erros.

Assim, o nosso projeto tem como objetivo facilitar e agilizar este processo de conferência e recepção de materiais aquando a sua chegada aos respectivos armazéns, uma vez que é oferecido à empresa uma aplicação *web* que permite, a partir de um código de barras referente a um produto ou encomenda validar com rigor e facilidade o produto recepcionado. Esta aplicação, ao fazer a leitura do código de barras do produto, permite visualizar todas as características do mesmo, sendo fácil a verificação da encomenda.

Tal como diz o ditado, *“Tempo é dinheiro”* e, desta forma o nosso projeto permite diminuir os tempos despendidos em cada recepção de encomendas aumentando o poder competitivo da empresa face ao mercado.

**2. User Stories**

A aplicação *web* desenvolvida é destinada aos funcionários de recepção de materiais em armazéns, no entanto, esta opera em conjunto com o Primavera ERP. A utilização da aplicação desenvolvida tem como objetivo facilitar e tornar mais rápida a gestão de entradas de novos artigos em stock, passando todo o controlo destas funcionalidades do ERP para um computador ou *Tablet* no local de recepção do armazém junto do funcionário.

Nesta secção encontram-se descritas todas as formas de interação com a aplicação desenvolvida do ponto de vista de um funcionário, na forma de *user stories*[2].

* Como funcionário de recepção de encomendas, quero validar uma encomenda para saber se deve ser recebida no meu armazém.
* Como funcionário de recepção de encomendas, quero validar um artigo para saber se deve ser recebido no meu armazém.
* Como funcionário de recepção de encomendas, quero validar um fornecedor para que saber se é um dos fornecedores da empresa.
* Como funcionário de recepção de encomendas, quero validar um armazém para saber se é um dos armazéns da empresa.
* Como funcionário de recepção de encomendas, quero pesquisar encomendas pelo código de barras para saber se a encomenda existe e está pendente.
* Como funcionário de recepção de encomendas, quero pesquisar artigos pelo código de barras para saber se estes estão encomendados e pendentes.
* Como funcionário de recepção de encomendas, quero pesquisar fornecedores pelo código de barras para saber as informações como nome, morada, NIF e código de barras destes.
* Como funcionário de recepção de encomendas, quero pesquisar armazéns pelo código de barras para saber as informações com nome, morada e código de barras destes.
* Como funcionário de recepção de encomendas, quero pesquisar encomendas por fornecedor para saber quais são aquelas a receber de um dado fornecedor”.
* Como funcionário de recepção de encomendas, quero pesquisar encomendas por artigos para saber a que encomenda pertence.
* Como funcionário de recepção de encomendas, quero pesquisar encomendas por armazém para saber quais as encomendas pendentes a receber num dado armazém.
* Como funcionário de recepção de encomendas, quero gerar uma guia de remessa dos artigos encomendados que recebi para declarar a recepção dos mesmos e atualizar os *stocks* do armazém.
* Como funcionário de recepção de encomendas, quero retificar a quantidade recebida de um tipo de artigo para atualizar os stocks em armazém corretamente.

**3. Solution Framework e Tecnologias Adotadas**

Para o desenvolvimento do projeto foram utilizadas diferentes ferramentas. Nesta secção do relatório pode ser consultada uma lista de todas elas, incluindo a sua utilidade no projeto final.

* Primavera Profissional ERP
* versão de educação 8.10 – plataforma Windows
* Sistema de Informação
* Primavera Profissional Administrador
* versão de educação 8.10 – plataforma Windows
* Administrador do Sistema de Informação
* Microsoft Visual Studio Express for Web 2013
* versão 12.0.30723.00, *update* 3 – plataforma Windows
* IDE para o desenvolvimento da API em linguagem *C#*
* Brackets [3]
* versão Realease 1.0, build 1.0.0-15191 – plataforma OS X
* Processador de texto utilizado no desenvolvimento *front end* para as linguagens *HTML5*, *CSS3* e *JavaScript*
* ngrok [4]
* versão 1.7 – plataforma Windows
* Exposição pública do *localhost* da máquina
* bwip-js [5]
* versão 0.5
* Escritor de Códigos de Barra em JavaScript
* IcoMoon [6]
* versão livre (*Free Version*)
* Ferramenta de Iconografia e Gestão de Ícons
* SourceTree [7]
* versão 1.6.11.0 – plataforma Windows;
* versão 1.9.0 – plataforma OS X.
* Controlador de versões do código do projeto

**4. Core Views**

Nesta seção, encontram-se os protótipos das diferentes interfaces criadas para a aplicação desenvolvida pelo grupo de trabalho. A Figura 1, representa a página inicial, onde cada funcionário deve inserir as suas credências de acesso para que possa utilizar a aplicação *web*. Após a autenticação, o funcionário é redirecionado para a página principal (Figura 2.), onde este pode pesquisar encomendas, artigos, fornecedores e armazéns. As figuras 3, 4, 5 e 6 ilustram as páginas de informação detalhada para cada um dos resultados pesquisados.

Na figura 7, encontra-se ilustrado um diagrama de fluxo entre as interfaces descritas anteriormente.

**5. Arquitetura Tecnológica e Funcional**

Especificação detalhada dos *webservices* criados em C# e a camada de interoperabilidade implementada:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo | Input | Output | Controlador | Verbo HTTP |
| Verificar as credenciais de acesso para um utilizador | Model.Login | HttpResponseMessage | Login | Post |
| Buscar informações de um artigo consoante o seu ID | String id | Model.Artigo | Artigos | Get |
| Buscar todos os artigos contidos na base de dados |  | IEnumerable<Model.Artigo> | Artigos | Get |
| Buscar todas as encomendas ainda não completamente concluídas |  | IEnumerable< Model.DocCompra> | DocCompra | Get |
| Atualizar uma encomenda com artigos recepcionados | Model. EncomendaRecepcionada | HttpResponseMessage | DocCompra | Post |
| Buscar informações de um fornecedor | String Id | Model. Fornecedor | Fornecedor | Get |
| Buscar informações de um armazém | String id | Model.Armazem | Armazem | Get |
| Procurar todas as encomendas, artigos, armazéns ou fornecedores por concordância de um atributo por uma *string* | String query | Model.Search | Search | Get |

Todos os URLs são definidos da seguinte forma: *hostname*/api/{ Controlador }/.

**6. Instruções de Utilização**

**6.1 Autenticação na Aplicação**

A autenticação na aplicação desenvolvida é individual para cada utilizador.

O conjunto de credenciais de acesso apresentadas em seguida, podem ser utilizadas para autenticação na aplicação:

* + - Utilizador: *funcionario*
* Palavra Chave: *funcionario1*

**6.2 Execução da Máquina Virtual**

Para executar o projeto na máquina virtual, devem ser executados os três atalhos presentes no Ambiente de Trabalho, a saber: *api*, *ngrok, app*. O primeiro permite abrir o projeto em C# no Microsoft Visual Studio Express for Web 2013, que deve ser executado em modo de Administrador, para que a sua execução seja autorizada através do URL http://sinf.ngrok.com. O segundo atalho, inicia a execução da ferramenta ngrok que disponibiliza, publicamente, o servidor local da máquina virtual (o *localhost*). Finalmente, o atalho *app* inicia a aplicação *web* desenvolvida. Neste momento, a aplicação *web* encontra-se totalmente operacional.

**7. Referências**

1. Acedido em dezembro de 2014; adaptado de:

http://www.britannica.com/EBchecked/topic/287895/information-system

1. Acedido em dezembro de 2014; Utilizada a estrutura de user stories enunciada no artigo do endereço:

https://www.scrumalliance.org/community/articles/2013/september/agile-user-stories.aspx

1. Acedido em novembro de 2014: http://brackets.io
2. Acedido em novembro de 2014: https://ngrok.com
3. Acedido em novembro de 2014: https://code.google.com/p/bwip-js/
4. Acedido em novembro de 2014: https://icomoon.io
5. Acedido em novembro de 2014: http://www.sourcetreeapp.com