



**UNIVERSIDADE DE ÉVORA**

**Colégio Luís António Verney**

DEPARTAMENTO DE MATEMÁTICA

**Implementação dum processo  
iterativo para otimização de stocks**

**Modelação Estatística e Análise de Dados**

Aluno: Bruno Jorge Pereira da Silva

Orientação: Jorge Manuel Azevedo Santos

José Miguel Gomes *Saias*

**Mestrado em Modelação Estatística e Análise de  
Dados**

Dissertação

Évora, 2015

*Esta dissertação não inclui as críticas e as sugestões feitas pelo júri*





**UNIVERSIDADE DE ÉVORA**

**Colégio Luís António Verney**

DEPARTAMENTO DE MATEMÁTICA

**Implementação dum processo  
iterativo para otimização de stocks**

**Modelação Estatística e Análise de Dados**

Aluno: Bruno Jorge Pereira da Silva

Orientação: Jorge Manuel Azevedo Santos

José Miguel Gomes *Saias*

**Mestrado em Modelação Estatística e Análise de  
Dados**

Dissertação

Évora, 2015

*Esta dissertação não inclui as críticas e as sugestões feitas pelo júri*



# Agradecimentos

Quero agradecer às muitas pessoas que deram um importante contributo para eu chegar a esta fase da minha vida académica e pessoal, ao final deste trabalho e do curso.

Agradeço ao Professor Jorge Manuel Azevedo Santos pela oportunidade que me deu de realizar esta dissertação assim como pela orientação e conselhos, indispensáveis e muito importantes, prestados ao longo da mesma.

Agradeço ao Professor *José Miguel Gomes Saias*, pela disponibilidade e apoio sempre demonstrados ao longo deste trabalho.

Em particular e muito especialmente quero agradecer aos meus pais, Albertino Silva e Maria Fernanda Silva, que me proporcionaram a frequência no ensino universitário e que sempre me apoiaram nas minhas decisões, nos bons e maus momentos pessoais e académicos.

A todos o meu sincero agradecimento.



# Resumo

A otimização global dum Sistema de stocks com penúria permitida e com tempo de reposição e procura aleatórias sob suposição de baixas probabilidades de rotura, conduz a um processo iterativo com consulta de várias tabelas.

Dada a morosidade desse processo de cálculo manual, avançou-se para a resolução desse problema na web.

Esta técnica consiste em encomendar sempre a mesma quantidade  $Q$  sempre que o nível de existências desça abaixo dum limiar  $M$  denominado ponto de encomenda.

O objetivo consiste em determinar os valores de  $Q$  e de  $M$  que minimizam a soma de 3 parcelas: Custos de Encomenda (minimizados encomendendo grandes quantidades); Custos de Posse (minimizados encomendendo pequenas quantidades e com baixos níveis de encomenda) e finalmente Custos de rotura (minimizados adoptando altos níveis de encomenda).

Verifica-se que existem conflitos entre os vários objetivos, pelo que se tem de resolver um problema de minimização não linear.

Palavras-chave

Otimização, Stocks, Probabilidade de Rotura, Fórmula de Wilson, Programação Matemática.



# Implementation of an Iterative Process For Inventory Optimization

## Abstract

The global optimization of an inventory system with shortages permitted model and with random lead time and random demand under the assumption of low probabilities of shortage leads to an iterative process with consultation of tables.

Given the slowness of this manual calculation process, we have made progress in solving this problem on the web.

This technique consists of always ordering the same amount  $Q$  when the inventory level falls below a calculated re-order threshold level  $M$ .

The objective is to determine the values of  $Q$  and  $M$  that minimize the three parts: Order Cost (minimized ordering large quantities); Holding costs (minimized ordering often small amounts) and finally shortage costs (minimized by adopting high re-order levels).

There are conflicts between the various objectives, so it has to be solved by a nonlinear minimization problem.

Key-word

Optimization, Stocks, Shortage Probability, Wilson Formula, Math Programming.



# Índice

Agradecimentos.....	5
Resumo.....	7
Abstract.....	9
Capítulo 1:.....	13
Introdução.....	13
1.1. Motivação.....	13
1.2. Objetivos do Trabalho.....	14
1.3. Metodologia.....	14
1.4. Estrutura do Relatório.....	15
Capítulo 2:.....	16
Conceitos e Enquadramento Teórico.....	16
2.1. Gestão de Stocks.....	16
2.1.1. Vantagens e Desvantagens na Constituição de Stocks.....	17
2.1.2. Objetivos dos Modelos de Gestão de Stocks.....	17
2.1.3. Componentes do Sistema de Gestão de Stocks.....	19
2.2. Modelos de Gestão de Stocks.....	24
2.2.1. Modelos Determinísticos.....	24
2.2.2. Modelos Estocásticos.....	32
Capítulo 3:.....	50
Desenvolvimento do Protótipo.....	50
3.1. Objetivos da Aplicação.....	51
3.2. Linguagem e Arquitetura.....	51
3.2.1. Para a Janela Java Swing.....	52
3.2.2. Para a Interface Web.....	52
3.3. Diagramas.....	52
3.3.1. Diagrama de Classes Janela Java Swing.....	52
3.3.2. Diagrama de Classes Interface Web.....	53
3.3.3. Diagrama de Páginas Interface Web.....	53
3.4. Classes, Métodos Importantes.....	54
3.4.1. Janela Java Swing.....	54
3.4.2. Interface Web.....	55
3.5. Requisitos.....	56
3.6. Hiperligações.....	57

3.6.1. Aplicação web .....	57
3.6.2. Executáveis.....	57
Capítulo 4: .....	58
Conclusões e Trabalho Futuro .....	58
Referências Bibliográficas.....	60
Anexos I: Manual do Utilizador (em português) .....	62
Anexo II: User Guide (in english) .....	86

# Capítulo 1:

## Introdução

### 1.1. Motivação

Podemos traduzir stock através do termo económico existência, provisão ou de forma mais simples, mercadoria armazenada que pode ter ou não uma grande rotatividade.

Quantas vezes nos dirigimos a uma loja e queremos comprar algum produto e as respostas que ouvimos são: “Desculpe mas já não temos!”, “Está esgotado!”, “Temos de encomendar! ”. Isto é o reflexo da má gestão dos stocks da empresa. Estas deveriam efectuar encomendas quando o stock de um determinado item chega a um número reduzido de unidades em stock, para evitar a falta do mesmo quando solicitado por um cliente. Pois assim melhoravam a capacidade de resposta relativamente às vendas.

Nos anos 50, desenvolveu-se no Japão um sistema de inventário de produção designado por Just-In-Time, que veio minimizar os stocks e os custos com o armazenamento.

Os stocks representam uma parte significativa do capital investido numa empresa. Se os produtos tiverem baixa rotatividade a empresa terá capital parado que poderia ser investido noutras áreas.

No geral, os custos associados ao armazenamento são muito elevados, por vezes chega a ser cerca de um quarto do valor do inventário. (CARVALHO, 2010). Portanto, se conseguirmos otimizar a gestão de stocks podemos reduzir os custos de armazenamento, evitando assim ter um elevado número de stocks e aumentando a competitividade da empresa.

Desta forma, se houver um controlo de forma a otimizar os níveis de stock e minimizar os custos com o armazenamento, a empresa aumentará os seus lucros. Esta é a principal razão para utilizar uma ferramenta de apoio à decisão na otimização de stocks. Caso os produtos tenham uma elevada

rotatividade a empresa terá de gerir o espaço alocado em armazém, com a otimização dos stocks isto já é mais fácil.

Em caso geral, na indústria, as empresas cada vez mais procuram investigar formas para melhorar os seus processos quer a nível de reposição quer a nível de distribuição.

As empresas têm de ter a noção dos produtos mais importantes e os menos importantes relativamente à sua rotatividade. Aqueles que têm uma maior rotatividade correspondem ao que têm maior retorno e os menos importantes a menores retornos.

## 1.2. Objetivos do Trabalho

O objetivo fundamental deste trabalho é desenvolver uma ferramenta que calcule iterativamente a quantidade ótima a encomendar utilizando modelos de otimização global dum sistema de stocks com penúria permitida e com tempo de reposição e procura aleatórias sob suposição de baixas probabilidades de rotura por ciclo ( $\alpha$ ).

Assim decidiu-se implementar uma ferramenta de apoio à decisão com dois layouts distintos, Website e Java Swing (versão portátil).

## 1.3. Metodologia

Hoje em dia com o desenvolvimento e crescimento das áreas de Tecnologias de Informação é possível desenvolver ferramentas como programas e websites de forma a dar ao utilizador uma melhor solução que auxilie na otimização de stocks.

Para concretizar os objetivos acima propostos, a metodologia utilizada incidiu fundamentalmente na modelação e codificação de um processo iterativo capaz de calcular a quantidade ótima a encomendar a partir do preenchimento dos seguintes parâmetros.

- Custo de Encomenda
- Custo de Posse
- Custo de Rotura

- Procura Média
- Variância da Procura durante o Tempo de Reposição

Para isto foi utilizada a plataforma NetBeans IDE 8.0.2, onde a linguagem de programação foi Java, para o desenvolvimento da versão Java Swing e Java com XHTML para o desenvolvimento do Website.

## 1.4. Estrutura do Relatório

Este relatório encontra-se dividido em seis capítulos:

No primeiro capítulo é feita uma exposição das ideias principais subjacentes a este trabalho, onde se identifica o âmbito do projecto, os objectivos propostos a atingir para a sua realização e a metodologia utilizada.

O segundo capítulo refere-se ao enquadramento teórico, no qual é apresentado o trabalho de pesquisa bibliográfica efetuado. Os temas abordados incidem sobre os modelos de gestão stocks, determinísticos e estocásticos, serão também apresentados algumas das componentes dos modelos de gestão de stocks. Também será feita uma apresentação muito breve dos conceitos de otimização global e otimização parcial.

O terceiro capítulo é dedicado ao desenvolvimento do protótipo. É iniciado com uma descrição dos requisitos do sistema, seguidos de alguns diagramas, muito poucos, dado que o sistema é simples e pequeno. Em termos gerais é um capítulo com uma abordagem prática que pretende expor com um fio condutor a sequência de procedimentos com o qual foi realizado.

E no quarto capítulo são apresentadas as conclusões do estudo, as sugestões efetuadas e também o balanço final do projeto, onde se apresentam contributos para uma melhor gestão de stocks com o objetivo de eliminar os problemas identificados.

## Capítulo 2:

# Conceitos e Enquadramento Teórico

As tecnologias têm sofrido grandes transformações a um ritmo bastante elevado. Da mesma forma, as empresas tendem a acompanhar a evolução tecnológica por forma a melhorar os seus processos produtivos. As tecnologias permitem às empresas partilharem informações sobre os seus negócios de forma a publicitá-los e também a troca de informação entre os vários departamentos, clientes e até empresas. Pode-se constatar que, as Tecnologias de Informação (TI) são uma área bastante importante para o sucesso das empresas.

## 2.1. Gestão de Stocks

Começamos por falar um pouco sobre a gestão de stocks. Mas primeiro tem de se perceber este conceito. Entende-se por stock uma reserva de materiais ou produtos que tem como objetivo facilitar a produção ou venda de produtos, conseguindo assim satisfazer a procura dos clientes (Costa, 2002). Os stocks representam, de um ponto de vista da economia, uma grande percentagem do capital imobilizado de uma empresa.

Um dos grandes desafios nesta área é perceber qual o nível de stock adequado para cada produto, de forma a evitar roturas. Por exemplo, no caso de alguns medicamentos não pode haver faltas, especialmente daqueles que possam causar danos irreparáveis, devido a negligências na gestão. Para manter um nível de stock aceitável, que dê resposta às necessidades, é necessário um controlo eficiente e a utilização de instrumentos para registo das informações e que facilitem o acompanhamento. Para calcular o tempo de reposição devem ser considerados os prazos necessários para a execução da compra.

## 2.1.1. Vantagens e Desvantagens na Constituição de Stocks

No *Quadro 1* serão apresentadas algumas das vantagens e desvantagens na constituição de stocks.

<b>Vantagens</b>
Assegurar o consumo regular de determinado produto;
Permitir reduzir o preço unitário na compra de uma maior quantidade (isto implica menores custos com transporte e descontos de quantidade);
Permite ter uma segurança no caso de haver atrasos na entrega por parte do fornecedor;
Permite que a aquisição de produtos não seja feita com muita frequência.
<b>Desvantagens</b>
Fragilidade de certos produtos (prazos de validade curtos;
Custos associados à manutenção de produtos que não foram consumidos;
Possíveis consequências, drásticas, no caso de se verificar rotura de determinados produtos (onde os custos de rotura dos mesmos são elevados).

*Quadro 1- Vantagens e Desvantagens na Constituição de Stocks*

## 2.1.2. Objetivos dos Modelos de Gestão de Stocks

As decisões relacionadas com a gestão de stocks referem-se à definição dos seguintes parâmetros de gestão:

- Quando devem ser encomendadas;
- Quanto encomendar de cada vez, isto é, a determinação da quantidade  $Q$  encomenda.

Em qualquer um dos problemas de decisão acima mencionados, a escolha de uma solução pressupõe a existência de um critério que permita comparar a atractividade ou eficácia dessas alternativas.

Na área de gestão de stocks têm sido propostos vários critérios que se poderão ajustar consoante a situação. Um destes critérios é o da *minimização de custos*, que é o mais utilizado. Podemos dizer que tirando situações inflacionárias ou de especulação, um sistema de gestão de stocks não é lucrativo per si. Por outro lado, este representa uma imobilização de recursos em actividades não reprodutivas, surgindo como um “mal necessário” para fazer face ao desajustamento dos processos de fornecimento e de procura. A *minimização de custos* pode ser considerada um critério aceitável, onde serão apresentadas as várias componentes do custo de um sistema de gestão de stocks no ponto 2.2.2.

Os modelos quantitativos de gestão de stocks permitem racionalizar as decisões tomadas adotando soluções de equilíbrio entre as alternativas:

- i) Que levam à acumulação de grandes stocks, com os consequentes custos de posse elevados;
- ii) Que conduzem sistematicamente a roturas de stocks com as consequências negativas de prestação de um mau serviço ao cliente e de potenciais perdas de vendas.

A parte mais significativa deste documento é dedicada aos modelos estocásticos de gestão de stocks aplicáveis aos artigos de consumo ao longo do tempo. Estes modelos são muito utilizados numa grande parte da análise e resolução na maioria dos problemas de gestão de stocks.

A existência de muitos artigos constitui um factor de escala que vai transformar pequenas economias, que se possam fazer pela gestão ótima de cada um, em grandes economias potenciais para a globalidade dos artigos em stock. Assim como traz também restrições de recursos globais, tais como espaço de armazenagem ou capital, que poderão não ser satisfeitas pelos parâmetros obtidos pela optimização individual de cada artigo.

A determinação dos parâmetros óptimos de gestão de cada artigo, que satisfazem as restrições globais, constitui um problema de optimização com restrições pouco utilizadas na prática (consultar Hadley, 1963). Normalmente o que as empresas fazem é impor valores máximos para as existências de cada

artigo, do tipo “não devem exceder 3 meses de consumo médio”, que globalmente satisfazem as restrições. Estes valores de existências máximas podem ser obtidos por simulação ou por otimização com restrições para amostras típicas de artigos.

### 2.1.3. Componentes do Sistema de Gestão de Stocks

Os componentes de um sistema de gestão de stocks têm uma grande importância na determinação dos parâmetros ótimos de gestão. Estes componentes são custos que influenciam a seleção dos parâmetros são apenas os que variam quando os parâmetros de gestão variam.

Os custos mais importantes são:

1. Custos de Aquisição;
2. Custos de Encomenda;
3. Custos de Posse;
4. Custos de Rotura.

Abaixo será apresentado um exemplo concreto e serão descritos mais pormenorizadamente estes custos. Constatar-se-á que é difícil a quantificação de todas as componentes do custo de forma exata e, por sua vez, é necessário fazer simplificações na sua utilização nos modelos desenvolvidos.

#### Exemplo 1:

A XPTO empresa que fabrica automóveis produz os seus próprios motores, que são utilizados na produção dos automóveis. Os automóveis são montados num sistema de linha de montagem contínua a uma taxa de 11.000 automóveis por mês, com um rádio por automóvel. Estes rádios são produzidos pela mesma empresa em lotes, porque não se justifica a criação de uma linha de montagem. Portanto os rádios são colocados em inventário até que sejam necessários para a montagem no automóvel na linha de produção.

A empresa está interessada em determinar quando produzir um lote de rádios e quantos rádios para produzir em cada lote.

De seguida serão apresentados vários custos que devem ser considerados:

- Cada vez que um lote é produzido, acrescenta um custo de instalação de 3.000 euros. Este custo inclui o custo de fabrico, custos administrativos, de manutenção de registos, entre outros. É de notar que a existência desse custo argumenta para a produção de rádios em grandes lotes.
- O custo de produção unitário de um rádio (excluindo o custo de instalação) é de 150 euros, e independente da dimensão do lote produzido. (Normalmente, o custo de produção unitário não tem necessariamente de ser constante e pode diminuir com a dimensão do lote).
- A produção de rádios em grandes lotes conduz a um grande inventário. O custo estimado para manter um rádio em stock é de 1,5 euros por mês. Este custo inclui o custo de capital empatado no inventário, como já foi explicado anteriormente. Uma vez que o dinheiro investido em stocks não pode ser investido noutras formas produtivas, desta forma, este custo do capital consiste no retorno perdido, pois os usos alternativos do dinheiro devem ser renunciados. Outros componentes do custo de rotura incluem o custo da locação do espaço de armazenamento, o custo do seguro contra a perda de inventário por incêndio, roubo ou vandalismo, os impostos com base no valor do inventário, e o custo de pessoal que supervisionam e protegem o inventário.
- A rotura de rádios surge ocasionalmente, e estima-se que cada rádio não disponível quando necessário na montagem custa 1,5 euros por mês. Este custo de rotura inclui um custo adicional na instalação dos rádios após o automóvel ser totalmente montado.

## 1. Custos de Aquisição

É o custo a pagar ao fornecedor do material. Esta quantia representa o *custo das unidades compradas*. Um caso frequente é aquele em que o custo unitário de um artigo,  $C_1$ , é constante e independente da quantidade encomendada. O custo de  $Q$  unidades será  $C_1 \cdot Q$ . Nem sempre esta relação é linear com a quantidade encomendada, embora seja geralmente uma aproximação satisfatória. Não será correto a utilização desta aproximação caso hajam descontos variáveis com a quantidade encomendada ou transportada.

## 2. Custos de Encomenda

Nestes são embutidos os custos administrativos dos serviços de compras que fazem a colocação e das encomendas (desde o custo das comunicações até ao “custo” da chefia que frequentemente chama a si a responsabilidade de autorizar a encomenda) e os custos de receção quantitativa, qualitativa e classificativa.

A parcela dos custos fixos de encomenda podem ser avaliados de forma grosseira, dividindo o custo dos serviços de compras e receção pelo número de encomendas colocadas anualmente, sendo frequentemente mais elevada do que os gestores apontam quando lhes é pedida uma estimativa.

Estes custos, que não dependem da quantidade encomendada, serão referidos como custos fixos da colocação de uma encomenda e representados por  $A$ .

Estes custos podem ser representados por uma função de  $c(z)$ . A forma mais simples desta função é dada por  $c \cdot z$ , ou seja, é aquele que é diretamente proporcional à quantidade encomendada, onde  $c$  representa o preço unitário pago. Outra hipótese comum é  $c(z)$  ser composto por duas partes: um termo que é diretamente proporcional à quantidade encomendada e um termo que é uma constante  $K$  para  $z$  positivo e é 0 para  $z = 0$ .

$$c(z) = \begin{cases} 0 & , se z = 0 \\ K + cz, & , se z > 0 \end{cases}$$

onde  $K$  corresponde aos custos de setup e  $c$  corresponde aos custos unitários.

Esta constante  $K$  inclui o custo administrativo de encomenda, ou, quando se produz, os custos envolvidos na criação desde o início da produção.

No *Exemplo 1*, os automóveis são produzidos e o custo de instalação para uma produção é 9.000 euros. Além disso, cada um dos rádios custa 150 euros, de modo que o custo de produção quando se dá uma ordem de produção de rádios  $z$  é dado por:

$$c(z) = 9.000 + 150z, \text{ para } z \geq 0$$

### 3. Custos de Posse

Nestes custos estão incluídos os custos monetários diretos tais como os custos de seguros, impostos, quebras, roubos, renda do armazém e outros custos de funcionamento do armazém tais como luz, mão-de-obra, guardas, etc.

A parcela mais importante não é um custo monetário direto, mas sim um custo de oportunidade. O seu valor é igual à maior taxa de rendibilidade que a empresa poderia obter em investimentos alternativos. Por se ter o capital investido em stocks perde-se a oportunidade de o investir noutras actividades. O potencial rendimento desse capital representa um custo no sistema de gestão de stocks.

Como todos estes custos são difíceis de avaliar com precisão, assume-se que os custos de posse de stock são iguais a uma percentagem fixa do investimento em stocks. Esta taxa que deve cobrir os custos do capital é quase sempre superior a 20% ao ano. (Tavares, Oliveira, Themido, Isabel, 1996)

Existe um outro custo, que ainda não foi mencionado, mas que normalmente se inclui nos custos de posse que são os custos de obsolescência. Este é o custo por unidade que deve ser eliminada, com perda, por se ter tornado obsoleta. O seu valor é em qualquer instante igual ao custo original da unidade mais o lucro que se poderia ter obtido desde a data de compra até a data de obsolescência, se os fundos utilizados na compra desse artigo tivessem um investimento alternativo, menos o valor de salvado.

Na maioria dos sistemas de gestão de stocks este custo é incluído no custo de posse, já que, de uma forma geral, quanto maior for o volume do stock médio maior será o risco de obsolescência, e, portanto, maior será este custo.

#### 4. Custos de Rotura

Ao considerar os custos de rotura deve ter-se em atenção duas situações:

- Onde o cliente aguarda pela vinda dos artigos não existentes e necessários à satisfação da sua encomenda. Esta situação é normalmente designada por “*carteira de encomendas*” ou “*cliente cativo*”;
- Onde, perante a rotura, o cliente desiste dos elementos pretendidos. Esta situação designa-se por “*vendas perdidas*”.

Os custos de rotura geralmente são difíceis de estimar, pois para além dos custos diretos envolve também os custos indiretos relacionados com a imagem comercial da empresa.

Na primeira situação, nos custos diretos estão incluídos os custos administrativos de manter encomendas pendentes e o eventual custo de imobilizações dos equipamentos ou de aquisição dos artigos com urgência.

Na segunda situação, ou seja, quando ocorre a não existência de stock representa a perda da encomenda, onde os custos associados são também difíceis de estimar, nestes incluem-se a perda de lucros futuros da venda de outros artigos por parte do utilizador por este vir a adquiri-los a outro fornecedor, ou por desencorajar outros potenciais utilizadores de irem abastecer-se ao sistema.

No *Exemplo 1*, o custo de posse é avaliado continuamente com um custo de 1,5 euros, por rádio, no inventário ao mês, de modo que o custo médio de manutenção por mês é de 1,5 euros vezes o número médio de rádios em inventário.

## 2.2. Modelos de Gestão de Stocks

Para uma escolha mais acertada do modelo mais adequado é necessário avaliar a existência ou não existência de aleatoriedade tanto do lado da oferta como no lado da procura.

No caso da oferta, no lado do fornecedor, se o prazo de entrega for fixo e sempre cumprido e se as quantidades entregues corresponderem sempre às quantidades encomendadas, então a oferta não tem aleatoriedade associada, e por isso, é considerada determinística. Pelo contrário, o fornecedor pode ter um prazo de entrega variável e não entregar sempre as quantidades encomendadas, sendo por isso considerado que a oferta tem um comportamento aleatório.

No caso da procura, no lado do cliente, ou do mercado, se as quantidades procuradas forem conhecidas, então pode-se afirmar que a procura é determinística; se a procura for variável, incerta, então tem-se um cenário de procura aleatória.

Desta forma, os modelos de gestão de stocks dividem-se em dois grandes grupos: modelos determinísticos e modelos estocásticos.

### 2.2.1. Modelos Determinísticos

Estes modelos são os mais simples e para os mesmos serem aplicados, a procura e a oferta não podem ter nenhum tipo de aleatoriedade, estes parâmetros devem ser conhecidos e constantes.

Neste tipo de modelos existem duas variáveis decisórias, *quando* e *quanto*, que se resumem apenas a uma pois estão relacionadas pela fórmula

$$T = \frac{Q}{r} .$$

Fixando-se o intervalo de tempo entre encomendas e a quantidade a encomendar, que será necessário para satisfazer a procura durante o período de tempo estipulado. Ou então, fixando-se a quantidade a encomendar e o intervalo entre encomendas, será igual ao tempo necessário ao consumo da quantidade encomendada. Esta condição pode ser representada como:

$$Q = T . r$$

onde,

$Q$  – Quantidade encomendada.

$T$  – Intervalo de tempo entre encomendas.

$r$  – Procura por unidade de tempo (determinística).

Em seguida será apresentada uma lista de fórmulas importantes utilizadas nos modelos determinísticos.

Notação matemática dos elementos:

$A$  – Custo da encomenda (euros)

$C_1$  – Custo de compra (euros/tonelada)

$C_2$  – Custo de posse (euros/tonelada)

$C_3$  – Custo de rotura (euros/tonelada)

$Q$  – Quantidade encomendada (toneladas)

$r$  – Procura por unidade de tempo (tonelada/ano)

$\tau$  – Tempo médio de reposição (dias)

$K$  – Custo por unidade de tempo (euros)

Quantidade a Encomendar:

$$Q = T \cdot r$$

Custo de Posse:

$$C_2 = \frac{Q}{2} \cdot T$$

Custo Total:

$$C_T = A + C_2 \cdot \frac{Q}{2}$$

Custo Total por Unidade de Tempo:

$$K = \frac{C_T}{T} = \frac{A}{T} + C_2 \cdot \frac{Q}{2}$$

por outro lado, se se substituir  $T = \frac{Q}{r}$  temos:

$$K = -\frac{A \cdot r}{Q} + C_2 \cdot \frac{Q}{2}$$

### 2.2.1.1. Reposição Instantânea; penúria não permitida

Este modelo é implementado quando a reposição dos produtos é instantânea e a quantidade de produto em *stock* não admite roturas, ou seja, a rotura de *stocks* no negócio em causa tem um impacto demasiado relevante.

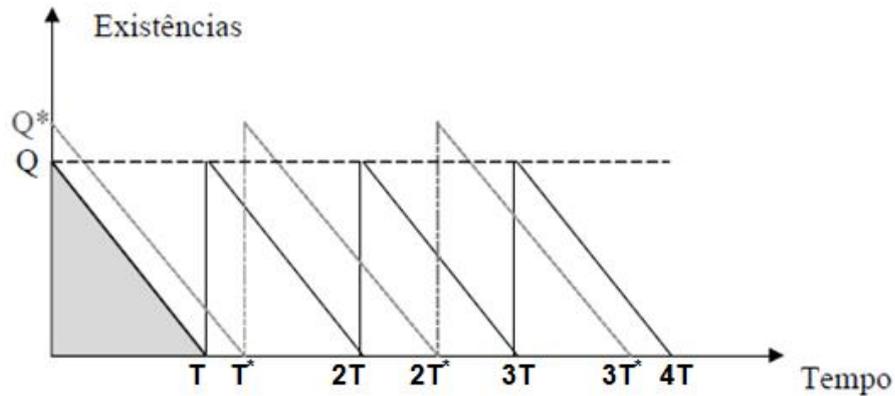


Figura 1 – Reposição Instantânea e Rotura Não Permitida

### 2.2.1.2. Reposição Instantânea; penúria permitida

Este modelo é implementado quando a reposição dos produtos é instantânea e a quantidade de produto em *stock* admite roturas, ou seja, a reposição dos produtos é considerada instantânea mas são admitidas roturas nos *stocks* dos produtos transaccionados.

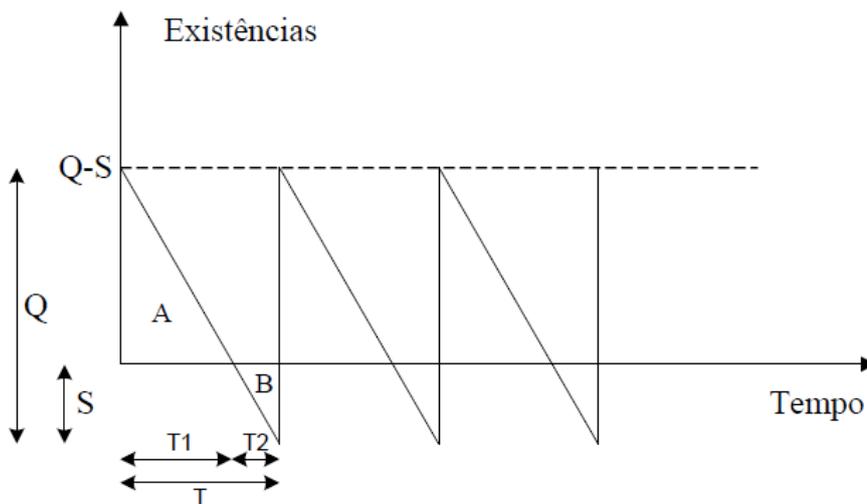


Figura 2 – Reposição Instantânea e Rotura Permitida

### 2.2.1.3. Reposição não Instantânea; penúria não permitida

Nos dois casos anteriores, a reposição do *stock* era instantânea, mas neste caso existe um período de reposição de *stock*, designado por  $T_1$ , ficando assim o modelo com uma taxa de reposição de *stock*, designado por  $p$ , que no final da reposição do *stock* atinge um nível máximo, designado por  $M$ .

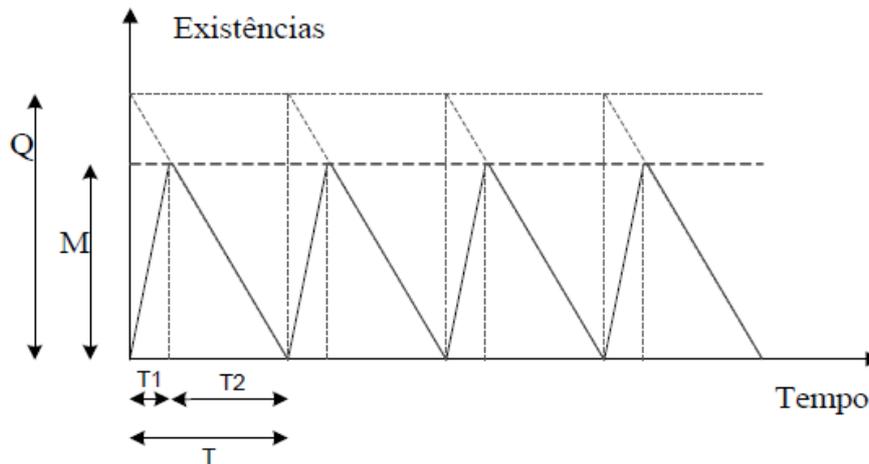


Figura 3 – Reposição não Instantânea e Rotura não Permitida

### 2.2.1.4. Reposição não Instantânea; penúria permitida

À semelhança do caso anterior a reposição não é instantânea e tem uma taxa de reposição. Como se pode observar na Figura 5, no caso deste modelo, tem-se quatro períodos de análise diferentes onde a rotura é permitida.

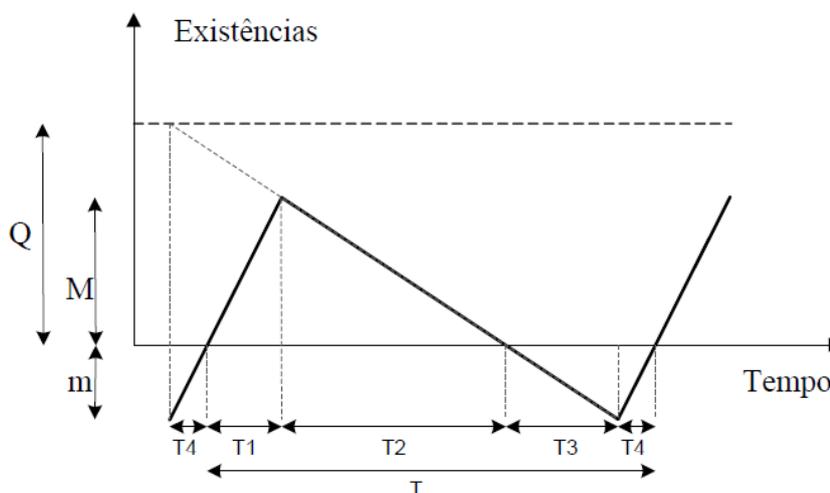
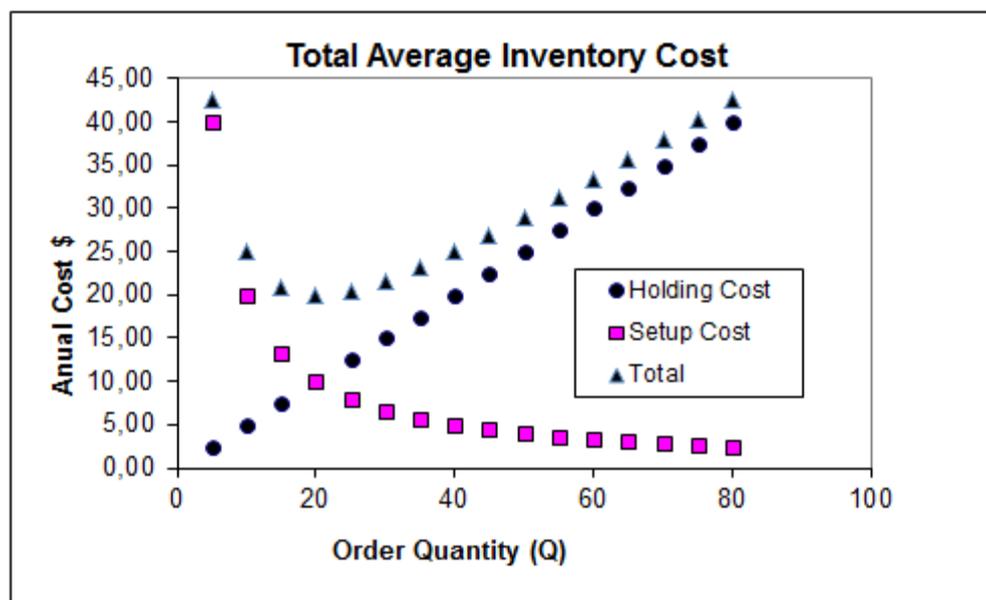


Figura 4 – Reposição não Instantânea e Rotura Permitida

### 2.2.1.5. Modelo da Quantidade Económica de Encomenda (QEE)

Este modelo tem como objetivo a minimização dos custos. Assim a resposta à questão “Quanto comprar?” é respondida por este modelo.

É de notar que, existe uma relação entre os custos acima mencionados, através da qual é possível determinar a quantidade ótima de encomenda (do inglês EOQ, Economical Order Quantity) ou lote ótimo,  $Q^*$ . Onde  $Q^*$  é o custo total mínimo. Como podemos observar na *Figura 5*.



*Figura 5 – Relação Entre os Custos*

A curva dos custos totais (por unidade de tempo), designada por  $K$ , em função de  $Q$ , bem como as curvas dos custos parcelares associados às encomendas e à posse do stock. Pode observar-se que:

i) A curva dos custos associados à colocação das encomendas (a primeira parcela da expressão) é uma função não linear decrescente com a quantidade a encomendar, isto é, quanto maior for o valor de  $Q$  menor será o número de encomendas a efectuar num mesmo período de tempo e, portanto, frequentemente se incorrer é na despesa fixa associada a cada encomenda;

ii) A curva dos custos de posse (a última parcela da expressão) é uma função linear crescente com a quantidade a encomendar, ou seja, quanto maior for o valor de Q, maior será também o nível médio de existências, e mais elevados serão os custos de manutenção desse stock;

iii) A curva dos custos totais, como o próprio nome indica é a soma das duas parcelas anteriores, dos custos de posse e dos custos de encomenda é uma função convexa. Pode dizer-se que esta expressão encontra um balanço harmonioso entre o investimento em stocks e despesas associadas à colocação e ao processamento de encomendas.

### 2.2.1.5.1. Fórmula de Wilson

Este tópico é dedicado apenas à dedução da fórmula do lote óptimo, também designada como fórmula de Wilson ou EOQ (Economical Order Quantity).

Para a dedução desta fórmula será necessária a fórmula do custo total por unidade de tempo (ver pág. 20).

#### Custo Total por Unidade de Tempo

$$K = -\frac{A \cdot r}{Q} + C_2 \cdot \frac{Q}{2}$$

Como foi referido anteriormente, o ponto de equilíbrio entre o custo de encomenda e o custo de posse de stock corresponde ao mínimo da função custo total. O mínimo da função CT é determinado através da derivação de CT em ordem a Q, igualar a zero, e resolver em função de Q.

Derivando K temos:

$$\frac{\partial K}{\partial Q} = -\frac{A \cdot r}{Q^2} + \frac{C_2}{2} = 0$$

$$\frac{A \cdot r}{Q^2} = \frac{C_2}{2}$$

Resolvendo em ordem a Q:

$$Q^2 = \frac{2.A.r}{C_2}$$

$$Q = \sqrt{\frac{2.A.r}{C_2}}$$

### 2.2.1.5.2. Modelo da QEE com Descontos de Quantidade

Nos modelos apresentados anteriormente, o preço era sempre independente da quantidade encomendada, de cada vez. Mas o que acontece muitas vezes é que os fornecedores oferecem descontos de quantidade para que se encomende mais quantidade de um dado material.

Caso existam descontos de quantidade, o custo de aquisição anual vai alterar-se, pois estes descontos incidem diretamente no custo de aquisição unitário dos artigos. Desta forma, o cálculo do custo total tem de incluir, para além do custo de encomenda anual e do custo de posse de stock anual, o custo de aquisição anual. Assim, o impacto do desconto de quantidades nestes três tipos de custos será o seguinte:

- Custo de aquisição anual que diminuirá dado que, o custo unitário de aquisição diminui, mantendo-se a quantidade a adquirir anualmente;
- Custo de encomenda anual que diminuirá dado que, para usufruir do desconto de quantidade, tem de se encomendar uma maior quantidade, o que implica que anualmente se coloquem menos encomendas ao fornecedor;
- Custo de posse de stock anual que aumentará dado que, a quantidade em stock será maior, porque se encomenda uma maior quantidade em cada encomenda.

Existem várias estruturas de descontos, entre elas uma bastante utilizada, a negociação de descontos ou bônus entre o cliente e o fornecedor para compras anuais de um determinado montante com a liberdade de encomendar conforme as necessidades. Outra estrutura também muito utilizada é a de descontos com três patamares onde cada gama de quantidade encomendada corresponde a um preço, como podemos observar abaixo.

$$Preço (P) = \begin{cases} P_1 & , 0 < Q < Q_1 \\ P_2 & , Q_1 \leq Q < Q_2 \\ P_3 & , Q \geq Q_2 \end{cases}$$

Para resolvermos este tipo de problemas temos de calcular  $Q^*$ , que é independente do preço e onde se admitem os custos de posse constantes.

No gráfico seguinte é apresentado um gráfico onde se relacionam os custos totais com desconto de quantidade (KTOT) em função da quantidade encomendada.

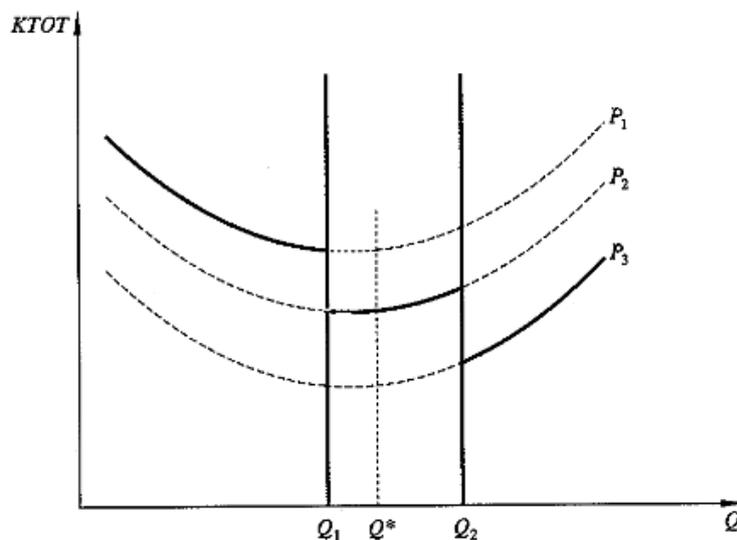


Figura 6 – Custo Total em Função dos Descontos de Quantidade

## 2.2.2. Modelos Estocásticos

Nos modelos anteriormente apresentados admitiu-se que todas as variáveis eram deterministicamente conhecidas. Pois essa não é uma boa solução porque não é isso que acontece na realidade.

Este trabalho incide sobre os modelos estocásticos onde se tem que os parâmetros procura e tempo de reposição são variáveis. Onde o tempo de reposição é o tempo que decorre desde, a colocação de uma encomenda para o abastecimento do stock, a entrada do material em armazém, para ser consumido.

Os modelos estocásticos têm como característica principal a procura ser desconhecida, ou seja, a procura tem um comportamento aleatório, incerto, que pode variar segundo uma distribuição estatística.

A incerteza presente neste tipo de modelos aumenta a complexidade da gestão de stocks, pois agora é necessário lidar com a possibilidade de existir rotura de stocks. Para se conseguir contornar este comportamento aleatório é necessário constituir um stock de segurança para compensar as flutuações superiores aos valores registados. Ao lidar com variáveis aleatórias, as flutuações que estas irão sofrer não se conseguem prever, apesar do stock de segurança conseguir fazer face a algumas dessas flutuações, mas não consegue na totalidade. Podemos constatar que quanto maior for o stock de segurança, maior é a probabilidade de ele conseguir fazer face às flutuações que não se conseguem prever; por outro lado, existe também a probabilidade de não conseguir, isto é, ao se falar de variáveis aleatórias é sinónimo de se falar em probabilidades.

Como acima referido, numa grande parte dos sistemas de gestão de stocks as variáveis decisórias são a quantidade a encomendar e o instante do tempo em que esta é colocada. No caso dos modelos determinísticos, a determinação do valor é simples e fixa e quando esta é calculada automaticamente é calculado o valor da outra. Mas não é o que acontece nos modelos estocásticos, onde os sistemas de gestão de stocks devem ser adaptados consoante as flutuações da procura variando o tempo entre encomendas ou variando a quantidade a encomendar. Estas duas possibilidades constituem dois grandes grupos de políticas de gestão de stocks

que serão sucintamente descritas neste tópico e que serão mais aprofundadas no tópico seguinte.

### 2.2.2.1. Probabilidade de Rotura por ciclo

Neste modelo pode-se considerar o ciclo dividido em duas partes:

i) A primeira parte considera-se desde o instante em que a encomenda chega até ao instante em que o stock-em-mão atinge o ponto de encomenda,  $M$ . Quando atingido  $M$  e este é positivo, nunca pode haver roturas durante esta parte do ciclo;

ii) A segunda parte do ciclo inicia-se no instante em que a encomenda é colocada, que termina quando esta chega. Assim, a duração desta fase é sempre igual ao tempo de reposição. No instante inicial (desta fase) o valor de stock-em-mão é igual ao valor de  $M$ . A existência de roturas vai depender do valor da procura durante a segunda parte do ciclo, isto é, durante o tempo de reposição.

Se a procura durante o tempo de reposição for superior a  $M$  haverá uma rotura de amplitude igual a procura verificada menos  $M$ . Se a procura durante o tempo de reposição for inferior a  $M$ , o stock-em-mão no instante em que a encomenda chega é positivo e não haverá rotura.

A quantidade em falta quando chega uma nova encomenda, pode ser calculada da seguinte forma:

$$\eta(x, M) = \begin{cases} 0, & \text{se } x \leq M \\ x - M, & \text{se } x > M \end{cases}$$

onde  $x$  representa a procura durante o tempo de reposição. Desta forma, o número esperado de unidades em falta por ciclo poderá ser calculado por:

$$\bar{\eta}(M) = \int_0^{+\infty} \eta(x, M) h(x) dx$$

$$\bar{\eta}(M) = \int_0^M h(x) dx + \int_M^{+\infty} (x + M)h(x) dx$$

$$\bar{\eta}(M) = \int_M^{+\infty} (x + M)h(x) dx$$

onde  $h(x)$  representa a função densidade de probabilidade da procura durante o tempo de reposição

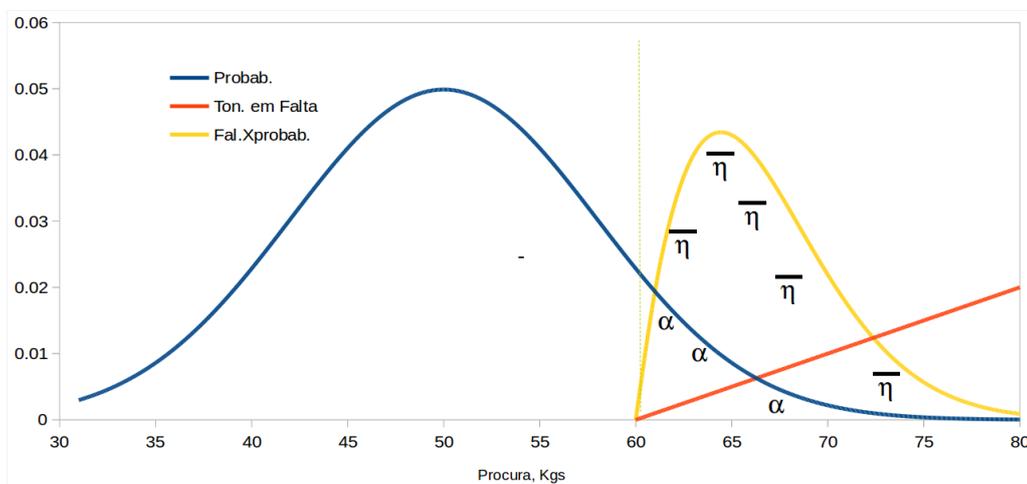
A probabilidade de rotura, designada por  $\alpha$ , é dada por:

$$P(x > M) = \int_M^{+\infty} h(x) dx$$

Esta probabilidade de rotura é dada pela área da Normal que tem início nas 60 toneladas (ver *Figura 7*). O número médio de unidades em falta é representado pela área debaixo da curva que se inicia nas 60 toneladas

O complementar da probabilidade de rotura ( $1 - \alpha$ ), normalmente é designado por nível de proteção, e é um indicador importante para medir a eficiência de uma política de gestão de stocks.

Em suma, pode-se dizer que, para esta política, a ocorrência de roturas depende do "comportamento" da procura durante o tempo de reposição. Para valores elevados da procura durante o tempo de reposição podem resultar de dois factores: de consumos excepcionais (muito elevados) ou de tempos de reposição mais longos do que se tinha previsto. Assim, como teremos oportunidade de demonstrar, o tempo de reposição vai desempenhar um papel muito importante no bom ou no mau funcionamento do sistema de gestão de stocks e no valor do investimento em stock necessário para proporcionar um bom nível de serviço.



*Figura 7 – Representação Gráfica da Distribuição Normal*

## 2.2.2.2. Relação entre M e Probabilidade de Rotura por ciclo

Considere-se um caso em que o tempo de reposição é constante e igual a uma semana. Neste caso a variabilidade da procura durante o tempo da entrega resulta apenas de alterações no consumo. Suponhamos que um levantamento dos consumos semanais do produto levou ao seguinte histograma representado na Figura abaixo. Neste caso o referido histograma dá-nos diretamente a distribuição  $h(x)$ .

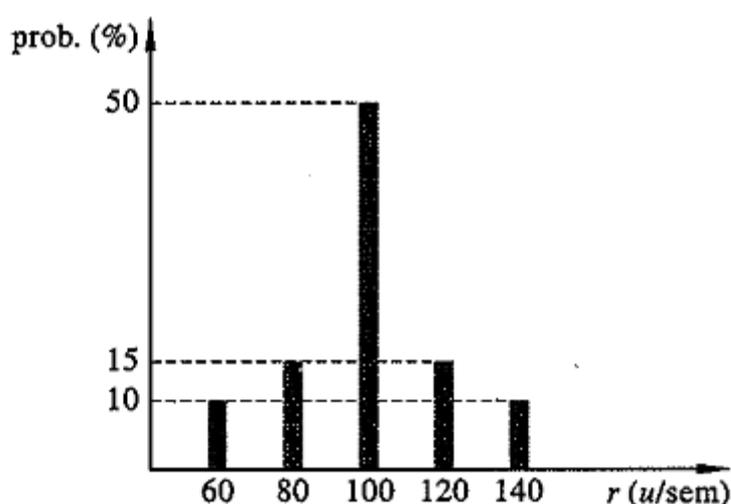


Figura 8 – Histograma da Procura Semanal

Ponto de Encomenda para M=100				
	Procura no tempo de reposição	Probabilidade (%)	Stock-em-mão no fim do ciclo (M - Procura)	Faltas
	60	10	40	0
	82	15	20	0
	100	50	0	0
	120	15	-20	20
	140	10	-40	40
Média	100		0	7

Quadro 2 – Ponto de Encomenda M=100

Se adotarmos um ponto de encomenda,  $M$ , de 100 unidades (ver quadro 2) haverá rotura quando a procura durante o tempo de reposição (1 semana) exceder as 100 unidades, isto é, quando for igual a 120 ou a 140 unidades. A probabilidade de rotura é, então, igual à soma das probabilidades destes dois acontecimentos, ou seja, 25%.

Quando a procura é igual a 120 unidades, 20 ficam por satisfazer; a probabilidade de isto acontecer é 15%. Quando a procura é igual a 140 unidades, 40 ficam por satisfazer; a probabilidade de isto ocorrer é igual a 10%. Como nas restantes 75% semanas não há rotura, ou seja, as faltas são iguais a zero, o valor médio das unidades em falta em cada ciclo,  $\bar{\eta}$ , é igual a:

$$\bar{\eta} = 0 * 0.75 + 20 * 0.15 + 40 * 0.1 = 7$$

O valor médio do stock de segurança,  $S$ , no fim do ciclo, imediatamente antes de a encomenda chegar é, por definição, o stock da segurança igual ao ponto de encomenda menos a procura média durante o tempo de reposição,  $\mu$ . Ou seja:

$$S = M - \mu$$

Para este caso,

$$S = 100 - 100 = 0$$

O stock de segurança obtido pela expressão anterior é igual à média do stock-em-mão no fim do ciclo, calculado no *Quadro 2*.

Como uma probabilidade de rotura por ciclo, designada por  $\alpha$ , 25% é uma probabilidade excessiva, na maioria das situações práticas, isto conduziria certamente a considerar valores de  $M$  mais elevados:

$M=100$	$\alpha=25\%$	$S=0$
$M=120$	$\alpha=10\%$	$S=20$
$M=140$	$\alpha=0\%$	$S=40$

### 2.2.2.3. Efeito do Aumento do Tempo de Reposição

Considere-se agora uma situação semelhante à anterior mas com um tempo de reposição fixo igual a duas semanas. Neste caso, para poder-se calcular o risco de rotura tem-se que determinar a distribuição da procura durante o tempo de reposição. Desta forma a procura durante o tempo de reposição,  $x$ , pode ser representada por:

$$x = \sum_{i=1}^L r_i$$

onde  $r_i$  é a procura no intervalo de tempo  $i$  e  $L$  o tempo de reposição.

1ª semana	2ª semana	1ª + 2ª semanas	Probabilidade
60	60	120	0.1 x 0.1= 0.01
	80	140	0,1 x 0.15= 0.015
	100	160	0,1 x0,5 =0.05
	120	180	0,1 x 0.15 = 0.015
	140	200	0,1 x 0.1 »0.01
80	60	140	0.15 x 0.1= 0.015
	80	160	0.15 x 0.15= 0.0225
	100	180	0.15 x0.5= 0.075
	120	200	0.15 x 0.15= 0.0225
	140	220	0.15 x 0,1= 0.015
100	60	160	0.5 x 0.1= 0.05
	80	180	0.5 x 0.15= 0.075
	100	200	0.5 x 0.5= 0.25
	120	220	0.5 x 0.15= 0.075
	140	240	0.5 x 0,1= 0.05
120	60	180	0.15 x 0.1= 0.015
	80	200	0.15 x 0.15= 0.0225
	100	220	0.15 x0.5= 0.075
	120	240	0.15 x 0.15= 0.225
	140	260	0.15 x 0,1= 0.015
140	60	200	0.1 x 0.1= 0.01
	80	220	0.1 x 0.15= 0.015
	100	240	0.1 x0.5= 0.05
	120	260	0.1 x 0.15= 0.015
	140	280	0.1 x0,1= 0.01

Quadro 3 – Calculo da Distribuição da Procura Para Duas Semanas

A função densidade de probabilidade agora será  $h(x)$ , que será condicionada da procura para um dado tempo de reposição, duas semanas. Admitindo que existe independência entre a procura em duas semanas consecutivas a determinação de  $h(x)$  pode ser determinada como apresenta o *Quadro 3*, onde se recorre a um método enumerativo de todas as situações possíveis.

Agrupando os valores, obtidos no quadro acima, da procura durante duas semanas obtemos os resultados que estão apresentados no *Quadro 4*:

Procura	Probabilidade de Ocorrência	
120	0.01	0.01
140	0.015 + 0.015	0.03
160	0.05 + 0.0225 + 0.05	0.1225
180	0.015 + 0.075 + 0.075 + 0.015	0.18
200	0.01 + 0.0225 + 0.25 + 0.0225 + 0.01	0.315
220	0.015 + 0.075 + 0.075 + 0.015	0.18
240	0.05 + 0.0225 + 0.05	0.1225
260	0.015 + 0.015	0.03
280	0.01	0.01
Procura média durante o tempo de reposição ( $\mu=200$ )		

*Quadro 4 - Procura média durante o tempo de reposição ( $\mu=200$ )*

Observe-se qual a probabilidade de rotura por ciclo,  $\alpha$ , e qual o *stock* de segurança para alguns valores possíveis do nível de encomenda, M:

M= 200	$\alpha = 34\%$	S= 0
M= 220	$\alpha = 16\%$	S= 20
M= 240	$\alpha = 4\%$	S= 40
M= 260	$\alpha = 1\%$	S= 60
M= 280	$\alpha = 0\%$	S= 80

Na Figura abaixo está representado o *stock* de segurança em função da probabilidade de rotura por ciclo para várias situações de tempo de reposição.

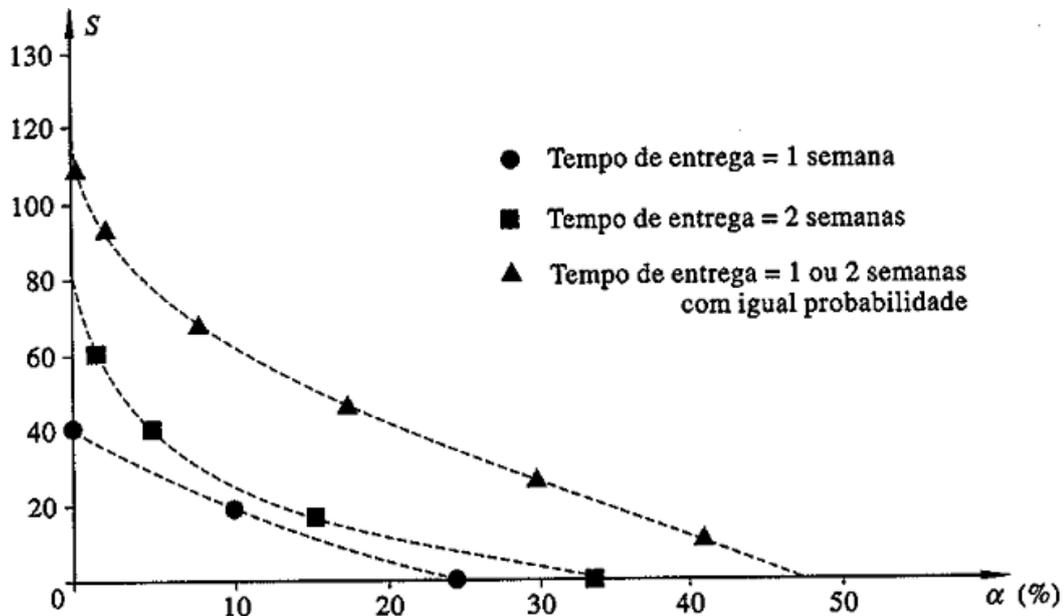


Figura 9 – Stock de Segurança em Função da Probabilidade de Rotura por Ciclo

Note-se que para valores de  $\alpha$  inferiores a 5% o *stock* de segurança para um tempo de reposição de duas semanas aumenta muito rapidamente, não havendo roturas para  $\alpha = 0\%$ , onde para os valores observados no tempo de reposição para duas semanas são o dobro dos valores observados para tempo de reposição para uma semana.

#### 2.2.2.4. Efeito da Variabilidade do Tempo de Reposição

A função densidade de probabilidade da procura durante o tempo de reposição varia consoante os casos e só uma análise de dados históricos, se existirem, pode permitir determiná-la. Em alternativa  $h(x)$  pode ser determinada a partir do conhecimento da função densidade de probabilidade do tempo de reposição,  $g(\tau)$ , e da função densidade de probabilidade condicionada da procura para um dado tempo de reposição,  $f(x; \tau)$ .

$$h(x) = \int_0^{+\infty} f(x; \tau)g(\tau)d\tau$$

Na prática nem sempre se consegue identificar  $g(\tau)$  e  $f(x; \tau)$  ou de combinar as duas distribuições.

Considere-se um caso em que o tempo de reposição também é variável, uma ou duas semanas com igual probabilidade. Conhecemos a distribuição condicional da procura para uma e duas semanas e a distribuição do tempo de reposição, assim pretendemos determinar a distribuição da procura durante o tempo de reposição.

Neste caso  $h(x)$  consegue-se calcular. Procedendo de modo idêntico ao utilizado anteriormente conhece-se os valores possíveis da procura, quando o tempo de reposição é uma semana, e quando o tempo de reposição é duas semanas. Como a probabilidade de ocorrência de cada um destes grupos de valores é 50%, admitindo independência, obtemos para a procura durante o tempo de reposição a distribuição apresentada no quadro abaixo:

Procura	Probabilidade de Ocorrência	
60	$0.5 \times 0.1$	0.05
80	$0.5 \times 0.15$	0.075
100	$0.5 \times 0.5$	0.25
120	$0.5 \times 0.15 + 0.5 \times 0.01$	0.08
140	$0.5 \times 0.1 + 0.5 \times 0.03$	0.065
160	$0.5 \times 0.1 + 0.5 \times 0.1225$	0.06125
180	$0.5 \times 0.1 + 0.5 \times 0.18$	0.09
200	$0.5 \times 0.1 + 0.5 \times 0.315$	0.01575
220	$0.5 \times 0.1 + 0.5 \times 0.18$	0.09
240	$0.5 \times 0.1 + 0.5 \times 0.1225$	0.06125
260	$0.5 \times 0.1 + 0.5 \times 0.03$	0.015
280	$0.5 \times 0.15 + 0.5 \times 0.01$	0.005

Procura média durante o tempo de reposição  
( $\mu=150$ )

Quadro 5 – Procura média durante o tempo de reposição ( $\mu=150$ )

Observe-se novamente os valores de  $\alpha$ , probabilidade de rotura por ciclo, e  $S$ , *stock* de segurança, para alguns valores possíveis de  $M$ :

M= 180	$\alpha = 33\%$	S= 30
M= 200	$\alpha = 17\%$	S= 50
M= 220	$\alpha = 8\%$	S= 70
M= 240	$\alpha = 2\%$	S= 90
M= 260	$\alpha = 0.5\%$	S= 110
M= 280	$\alpha = 0\%$	S= 130

Para este caso, os *stocks* de segurança são muito mais elevados do que as situações anteriores. Por exemplo, se se pretender garantir uma probabilidade de rotura por ciclo inferior a 5%, ter-se-ia de constituir um *stock* de segurança  $S = 30$  unidades para o tempo de reposição fosse uma semana,  $S = 40$ , se o tempo de reposição fosse igual a duas semanas, e  $S = 80$ , se o tempo de reposição fosse variável.

Neste exemplo podemos constatar o efeito do tempo de reposição sobre o *stock* de segurança e demonstra a razão pela qual as empresas preferem fornecedores e distribuidores com tempos de reposição curtos e pouco variáveis, estando dispostas a pagar preços mais elevados por serviços com essas características, que lhes permitem diminuir os investimentos em *stock*.

Muitas empresas conseguiram mesmo abolir os *stocks* de componentes e matérias-primas, já que tem fornecedores que lhes garantem a entrega no preciso momento em que o lote é necessário na linha de montagem ou de produção. Esta situação, altamente desejável em termos económicos, só é possível com fornecedores que ofereçam confiança ilimitada quer em termos de prazo de entrega quer em termos de qualidade.

### 2.2.2.5. Modelação da Procura Durante o Tempo de Reposição

Foi apresentado no ponto anterior que para a determinação do *stock* de segurança e do nível de proteção é indispensável conhecer as características estatísticas da procura durante o tempo de reposição. Foi dito acima que a determinação de  $h(x)$  resulta da combinação das distribuições da procura e do tempo de reposição. Serão apresentadas de seguida expressões genéricas para os parâmetros de  $h(x)$ , válidos para qualquer distribuição matemática.

Observe-se o caso em que o tempo de reposição é constante e igual a  $L$ . Nesse caso,  $x$  é dado pela expressão anterior, onde  $r_i$  é a procura, aleatória, na unidade de tempo.

Utilizando o valor esperado de  $x$  tem-se:

$$\mu = E(x) = E(\sum_{i=1}^L r_i)$$

$$\mu = \sum_{i=1}^L E(r_i)$$

$$\mu = \sum_{i=1}^L \bar{r}$$

$$\mu = L \bar{r}$$

onde  $\bar{r}$  corresponde à procura média na unidade de tempo. Admitindo-se que a procura é independente de unidade de tempo para unidade de tempo a variância de  $x$ , pode ser calculada por

$$\sigma^2 = VAR(x) = VAR(\sum_{i=1}^L r_i)$$

$$\sigma^2 = \sum_{i=1}^L VAR(r_i)$$

$$\sigma^2 = \sum_{i=1}^L \sigma_r^2$$

$$\sigma^2 = L \sigma_r^2$$

onde  $\sigma^2$  representa a variância da procura na unidade de tempo.

Pode-se mostrar que, quando  $x$  e  $L$  são ambos variáveis, a média e a variância da procura durante o tempo de reposição são dadas pelas

expressões seguintes para quaisquer distribuições da procura e do tempo de reposição.

$$\mu = \bar{t} \bar{r}$$

$$\sigma^2 = \bar{t} \sigma_r^2 + \sigma_t^2 \bar{r}$$

onde  $\bar{t}$  e  $\sigma_t^2$  são respetivamente a média e a variância do tempo de reposição, e  $\bar{r}$  e  $\sigma_r^2$ , são respetivamente a média e a variância da procura por unidade de tempo. Para o caso de a procura por unidade de tempo ser fixa e igual a  $r$ , mas com o tempo de reposição variável,  $\mu$  e  $\sigma$  podem ser determinados a partir das duas fórmulas apresentadas acima, bastando tomar:

$$\bar{r} = r \quad \text{e} \quad \sigma_r^2 = 0$$

Geralmente os modelos clássicos assumem para  $h(x)$  uma distribuição Normal, apesar de, em casos especiais por vezes se assumam outras distribuições, como a distribuição Gama, que se ajusta melhor as situações estudadas.

Se o tempo de reposição for suficientemente longo, poderá ser evocado o Teorema do Limite Central para mostrar que a distribuição de  $x$  terá uma aproximação à distribuição Normal. Quando se adota para  $h(x)$  uma distribuição Normal, de parâmetros  $\mu$  e  $\sigma$ , é usual exprimir o *stock* de segurança como:

$$S = Z_\alpha \sigma$$

onde  $Z_\alpha$  é chamado de factor de segurança. Assim,

$$\alpha = 1 - \Phi\left(\frac{M - \mu}{\sigma}\right) = 1 - \Phi(Z_\alpha)$$

onde  $\Phi(Z_\alpha)$  representa a probabilidade de a variável normal reduzida tomar valores menores ou iguais a  $Z_\alpha$ . No caso de  $h(x)$  seguir uma distribuição normal também o cálculo do número médio de unidades em falta por ciclo fica facilitado:

$$\int_M^{+\infty} (x - M) h(x) dx = \sigma \xi \left( \frac{M - \mu}{\sigma} \right) = \sigma \xi(Z_\alpha)$$

onde  $\xi(Z_\alpha)$  representa a função de perdas normal que se encontra tabelada.

### 2.2.2.6. Modelo de Revisão Contínua ou Política do Nível de Encomenda

O Modelo de Revisão Contínua ou do nível de encomenda é uma adaptação do modelo da quantidade económica de encomenda onde para intervalos fixos tempo entre encomendas a procura e/ou oferta são aleatórios. À semelhança do modelo de quantidade económica de encomenda, este difere pela existência do stock de segurança. Este tem uma monitorização contínua (constante) dos níveis de stock, designado por  $M$ . Esta revisão é importante, pois quando o nível de stock atinge uma quantidade pré-definida, designada por ponto de encomenda, é efetuar uma encomenda para o fornecedor. Se a encomenda não for feita assim que o nível de stock atinge o ponto de encomenda, o risco de rotura aumenta. Melhor explicando entre os vários pontos de revisão há sempre uma probabilidade de haver rotura de stock, pois o tempo de reposição em tempo útil é desconhecido, isto é, depende da capacidade de resposta do fornecedor perante as encomendas. Assim, os pedidos dos clientes que procurem o produto poderá não ser satisfeita

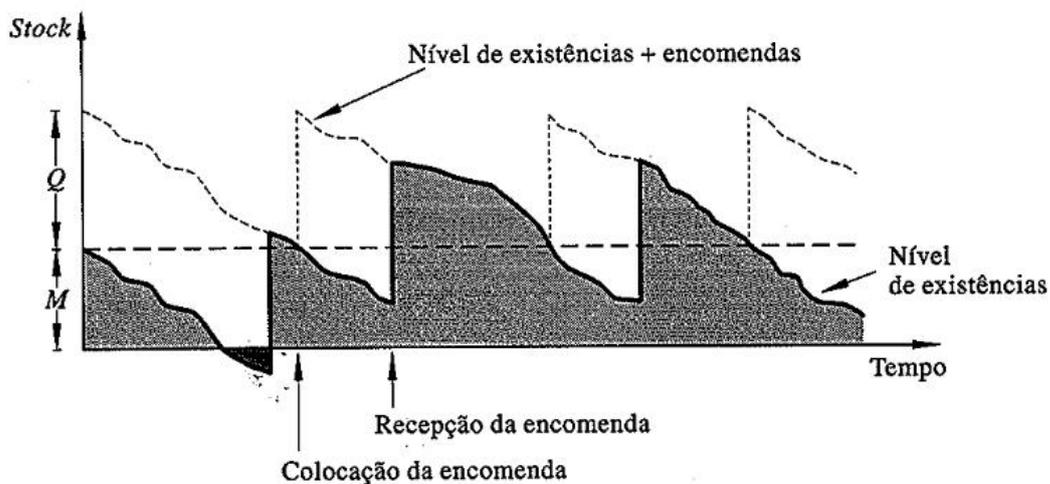


Figura 10 – Modelo de Revisão Contínua ou Política do Nível de Encomenda

Neste modelo, a quantidade a encomendar é fixa, designada por  $Q$ , mas o período entre encomendas é variável, isto é, depende do ritmo da procura no período entre encomendas. Como a procura e o prazo de entrega são variáveis, existe a possibilidade de rotura. Se se dividir o ciclo de encomenda em duas partes, quando a quantidade em stock é superior ao ponto de encomenda e quando a quantidade em stock é inferior ao ponto de encomenda), só existe probabilidade de rotura na segunda parte do ciclo, que corresponde ao tempo de entrega por parte do fornecedor. Só existirá rotura caso a procura durante o prazo de entrega por parte do fornecedor for superior ao ponto de encomenda. Como a procura durante o prazo de entrega é uma variável aleatória, é necessário identificar qual o tipo de distribuição estatística que essa variável segue e os parâmetros associados à mesma.

Considere-se um modelo de revisão contínua ou política de revisão cíclica com um prazo de entrega fixo por parte do fornecedor e que a procura (para este prazo de entrega) que segue aproximadamente uma distribuição estatística Normal. Que se pode representar por:

Por consequência, a probabilidade de haver procura durante o prazo de entrega ser menor ou igual ao ponto de encomenda definido, corresponderá ao nível de serviço que é a probabilidade de conseguir satisfazer as encomendas na quantidade e momento solicitados.

O stock de segurança é um parâmetro que já foi introduzido, o seu cálculo varia consoante a política de reaprovisionamento utilizada. No caso da revisão cíclica é calculado através do desvio padrão da procura,  $\sigma$ , e do nível de serviço ( $ns$ ) respectivo,  $Z$ .

Podem-se mostrar as relações necessárias à optimização dos stocks utilizando este modelo. O nível máximo,  $M$ , para o qual são efectuadas as encomendas é determinado somando o produto do tempo de entrega pela procura por unidade de tempo com o stock de segurança. Para determinar a quantidade económica,  $Q^*$ , pode ser utilizada a fórmula de Wilson (pág. 26).

### 2.2.2.7. Modelo de Revisão Periódica ou Política de Revisão Cíclica

No Modelo de Revisão Periódica ou Política de Revisão Cíclica, o dia de colocação de uma encomenda é pré-definido através de negociações com o fornecedor ou programação interna, onde o período entre encomendas é fixo (semanal, quinzenal, mensal, entre outras).

No dia estipulado para a colocação da encomenda, compara-se o stock existente e o stock necessário para o próximo período, onde a quantidade a encomendar corresponderá à diferença entre estes dois valores. O modelo designa-se por Revisão Periódica ou Política de Revisão Cíclica, pois os níveis de stock são revistos periodicamente, ou seja, de  $x$  em  $x$  tempo.

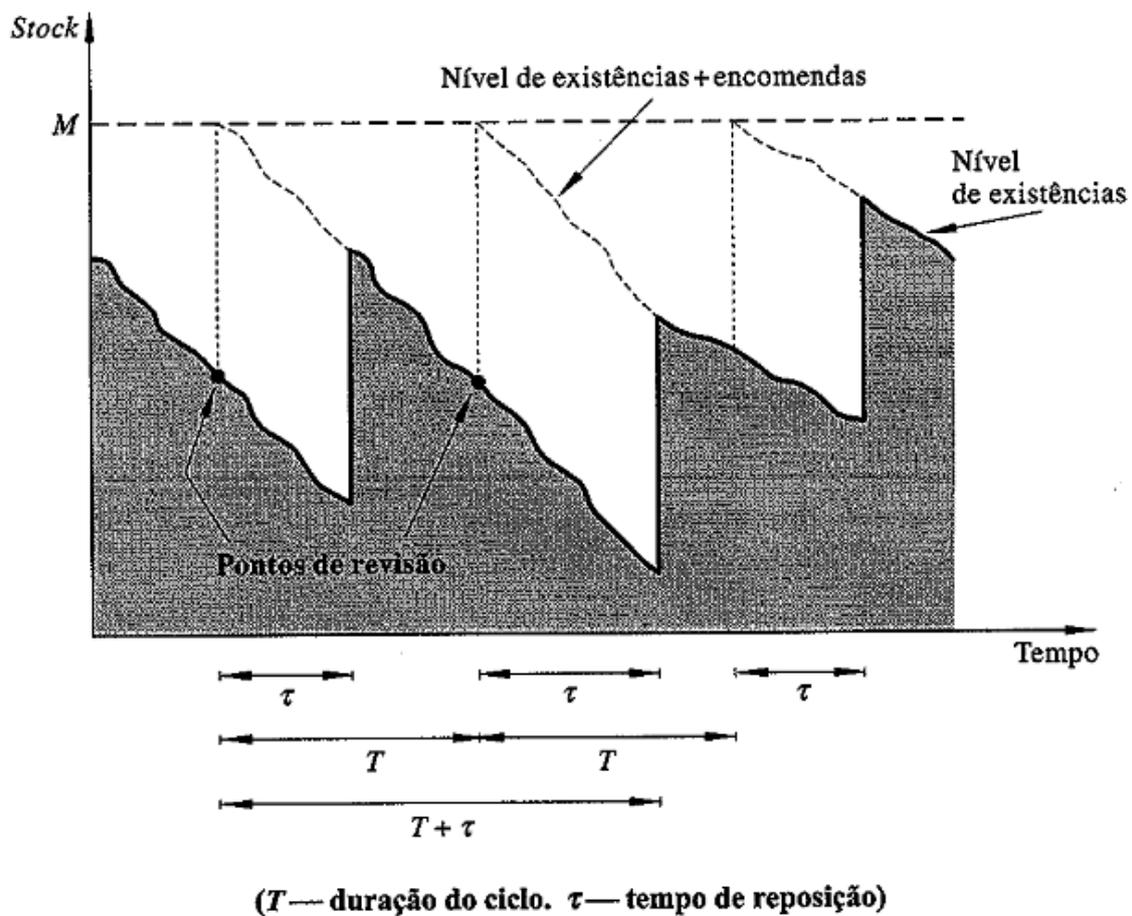


Figura 11 – Modelo de Revisão Periódica ou Política do Nível de Encomenda

## 2.2.2.8. Revisão Cíclica e Nível de Encomenda – Vantagens e Desvantagens

Na *Tabela 1* são apresentadas algumas das vantagens e desvantagens da utilização da política de Revisão Cíclica e da política do Nível de Encomenda.

	<b>Revisão Cíclica</b>	<b>Nível de Encomenda</b>
<b>Vantagens</b>	<ul style="list-style-type: none"><li>➤ Encomendas colocadas a intervalos fixos de tempo;</li><li>➤ Agregação de Encomendas.</li></ul>	<ul style="list-style-type: none"><li>➤ Encomendas de dimensão fixa;</li><li>➤ Aplica-se tanto a casos de produção como de revenda.</li></ul>
<b>Desvantagens</b>	<ul style="list-style-type: none"><li>➤ Risco de rotura de stocks entre pontos de revisão;</li><li>➤ Custos de armazenagem mais elevados assim como espaço alocado.</li></ul>	<ul style="list-style-type: none"><li>➤ Necessário conhecimento contínuo do sistema.</li></ul>

*Tabela 1 – Revisão Cíclica e Nível de Encomenda: Vantagens e Desvantagens*

### 2.2.2.9. Otimização Global

Quando se apresentou os modelos determinísticos, constatou-se que são modelos simples que visam a determinação dos valores das variáveis decisórias que minimizam os custos. No caso da otimização global procura-se caracterizar o comportamento do sistema contabilizando a aleatoriedade de certos factores.

Os problemas de optimizacao em modelos estocásticos são bastante mais complexos do que nos casos determinísticos, sendo este o foco deste estudo. Será apresentado apenas o caso da optimizacao global para a política do nível de encomenda que servira para exemplificar o tipo de metodologia a desenvolver nestas situações.

Considere-se a politica do nível de encomenda, admitindo que pode ocorrer rotura, que os pedidos não satisfeitos imediatamente não são perdidos, ficando em carteira. Admita-se ainda que existe um custo de rotura proporcional a quantidade em falta, mas independente do tempo de carência (representaremos por  $C'_3$  o custo de rotura por unidade em falta).

Como vimos acima, os custos de funcionamento podem ser decompostos em três grupos (encomenda, posse e rotura), que serão apresentados separadamente:

a) Custos de Encomenda:  $A$

b) Custos de Posse: por definição, o nível esperado de existências imediatamente antes de uma nova encomenda chegar é designado por stock de segurança, designado por  $S$ . Imediatamente depois de a encomenda chegar, o nível de existências passa a ser  $Q + S$ . Assim, se a chegada de uma nova encomenda for tomada como o inicio de um ciclo, as existências, em termos médios, são  $Q + S$  no inicio do ciclo, e  $S$  no final. O custo de pose é representado por  $C_2$ .

c) Custos de Rotura: O custo esperado de rotura por ciclo será o produto de  $C'_3$  pelo número médio de unidades em falta. O custo de rotura é representado por  $C_3$ .

Procedimento para a otimização global:

1) Começar com um valor de  $Q$  dado pela expressão do lote ótimo determinístico, fazendo  $r = \bar{r}$ ;

$$Q^* = \sqrt{\frac{2 \cdot A \cdot r}{C_2}}$$

2) Utilizar a seguinte expressão para determinar o valor de  $M$  correspondente a  $Q$ ;

$$\int_M^{+\infty} h(x) dx = \frac{C_2 Q^*}{C'_3 \bar{r}}$$

3) Utilizar a expressão abaixo, com o valor de  $M$  encontrado no passo anterior, para determinar um novo valor de  $Q$ ;

$$Q^* = \sqrt{\frac{2 r (A + C'_3 \int_M^{+\infty} (x - M^*) h(x) dx)}{C_2}}$$

4) Voltar a 2).

Este procedimento iterativo converge rapidamente, em geral, não será necessário ir além da terceira iteração para determinar uma boa aproximação de  $Q^*$  e  $M^*$ .

# Capítulo 3:

## Desenvolvimento do Protótipo

Este capítulo tem como objetivo dar a conhecer todos os aspetos técnicos da aplicação desenvolvida reunindo informações dispostas de forma sistematizada, criteriosa e segmentada para atuar como instrumento gerencial facilitando a compreensão da estrutura e funcionamento do Software.

De um modo geral a aplicação consiste em implementar um processo iterativo para a otimização de stocks com uma interface Web e uma interface Java Swing. Para criar a estrutura da aplicação foi utilizada uma ferramenta de ambiente de desenvolvimento integrado gratuita e de código livre, NetBeans, utilizando a linguagem Java, pois trata-se de uma linguagem orientada por objetos e muito utilizada nos dias de hoje. Para além disto tem bastante apoio a nível de livros, bem como de fóruns e tutoriais ao nível da Web, o que facilitou bastante, além disso é a linguagem em que me sentia mais à vontade. A nível da interface Java Swing utilizou-se a linguagem Java. Para a interface WEB reutilizou-se a classe onde são efetuados os cálculos (API), na linguagem Java, e para a parte visual propriamente dita utilizou-se a linguagem XHTML.

O modelo utilizado de desenvolvimento foi um misto de Modelo Incremental e Modelo por Protótipo, onde a partir do conjunto de requisitos se desenvolveu um pequeno protótipo e iterativamente se alcançou evoluções subsequentes das versões até o sistema todo estar implementado.

No final deste documento encontram-se, em anexo, dois manuais de suporte à aplicação, Manual do Utilizador (em português) localizado no Anexo I e User Guide (in english) localizado no Anexo II.

Este projecto foi dividido nas seguintes fases:

- Análise e especificação de requisitos
- Desenho
- Desenvolvimento
- Verificação e Validação

- Testes

Onde:

Na fase de Análise ou Identificação dos requisitos correspondeu à recolha dos dados e identificação das necessidades/exigências do utilizador para solucionar um problema e alcançar seus objetivos.

Na fase de Especificação dos requisitos correspondeu à tarefa de descrever precisamente como o Software irá funcionar.

Na fase de Desenho correspondeu a desenhar os diagramas, modelos e estrutura do sistema.

Na fase de Verificação e Validação correspondeu à verificação é a avaliação do sistema para determinar se o mesmo satisfaz as especificações definidas no início do projeto (é uma avaliação interna); validação é a avaliação para determinar se o sistema atende às necessidades do cliente.

Na fase de Testes correspondeu à investigação do Software com o objetivo de fornecer informações sobre a qualidade do mesmo.

Na fase de Documentação correspondeu à documentação interna do projeto é importante para propósitos de futuras manutenções, aprimoramentos, criação de novos módulos ou mesmo para integração com outros sistemas.

## 3.1. Objetivos da Aplicação

A aplicação consiste em implementar um processo iterativo para a otimização de stocks com uma interface Web e uma interface Java Swing apresentadas em dois idiomas, português e inglês.

## 3.2. Linguagem e Arquitetura

Optou-se pela ferramenta NetBeans, pois é uma ferramenta de desenvolvimento integrado gratuita e de código livre, para além disso permite

desenvolver Software em diversas linguagens entre elas o Java, HTML, XHTML. Além disso, é um ambiente que já me era familiar.

### 3.2.1. Para a Janela Java Swing

A nível do Layout e aparência visual da aplicação optou-se pela utilização de Java Swing no NetBeans.

### 3.2.2. Para a Interface Web

A nível do Layout e aparência visual da aplicação foi construída uma aplicação web baseada na tecnologia JavaServer Faces, alojada num servidor aplicativo Apache Tomcat no NetBeans.

## 3.3. Diagramas

### 3.3.1. Diagrama de Classes Janela Java Swing

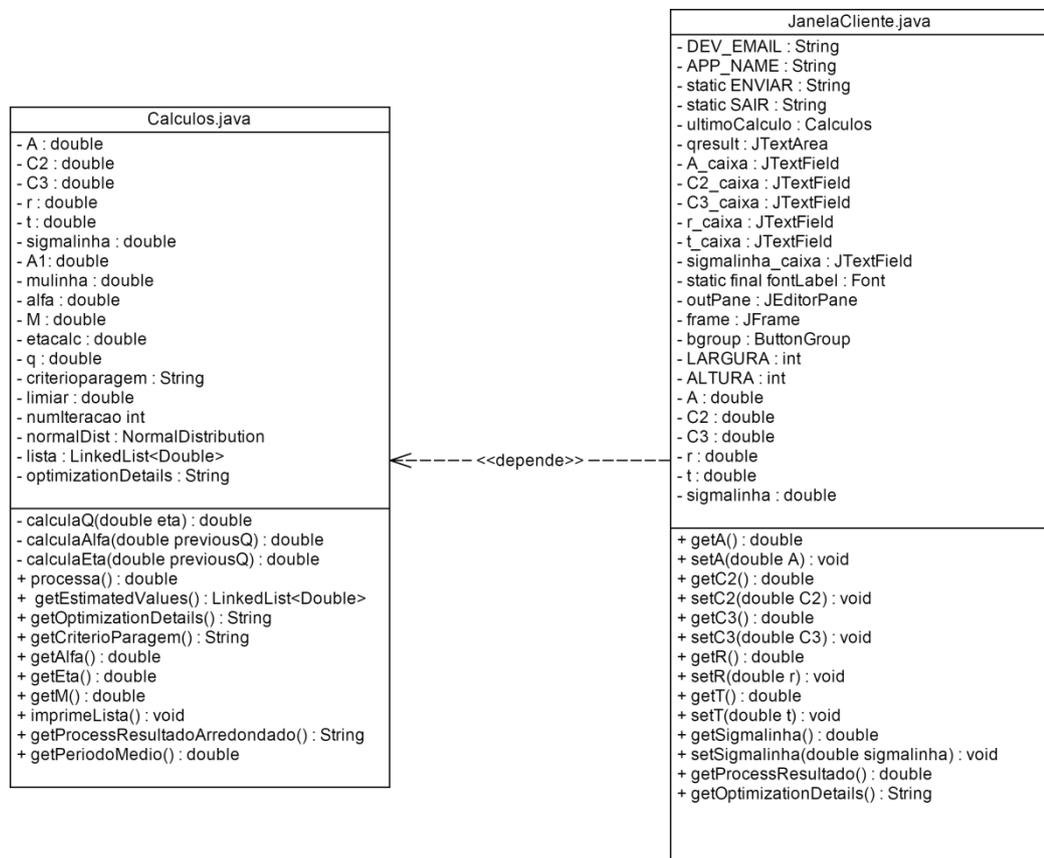


Figura 12 – Diagrama de Classes Janela Java Swing

### 3.3.2. Diagrama de Classes Interface Web

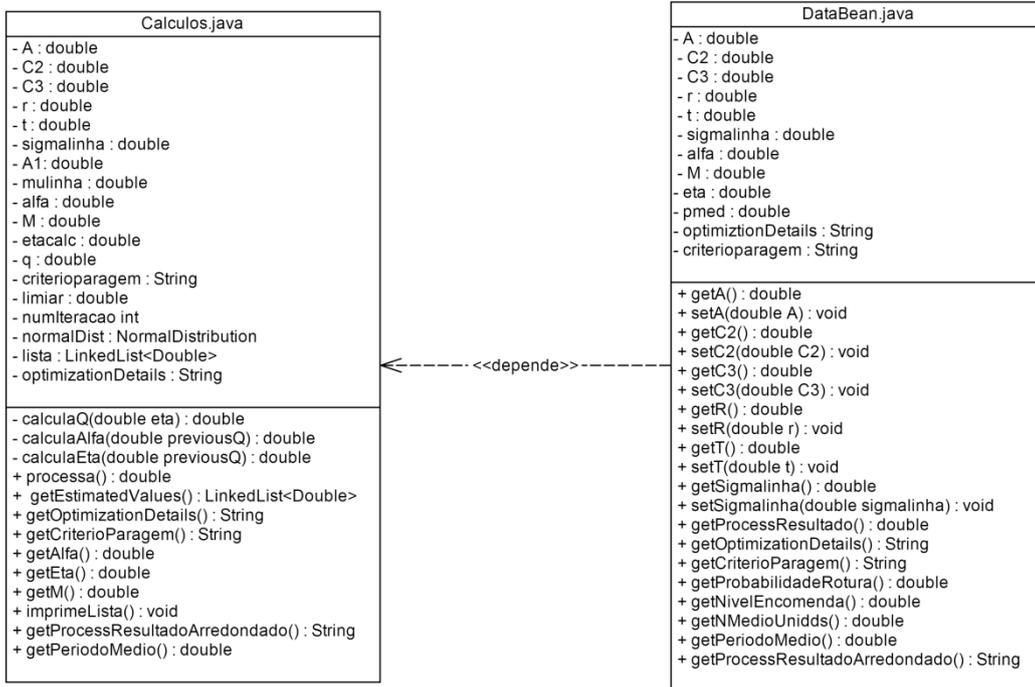


Figura 13 – Diagrama de Classes Interface Web

### 3.3.3. Diagrama de Páginas Interface Web

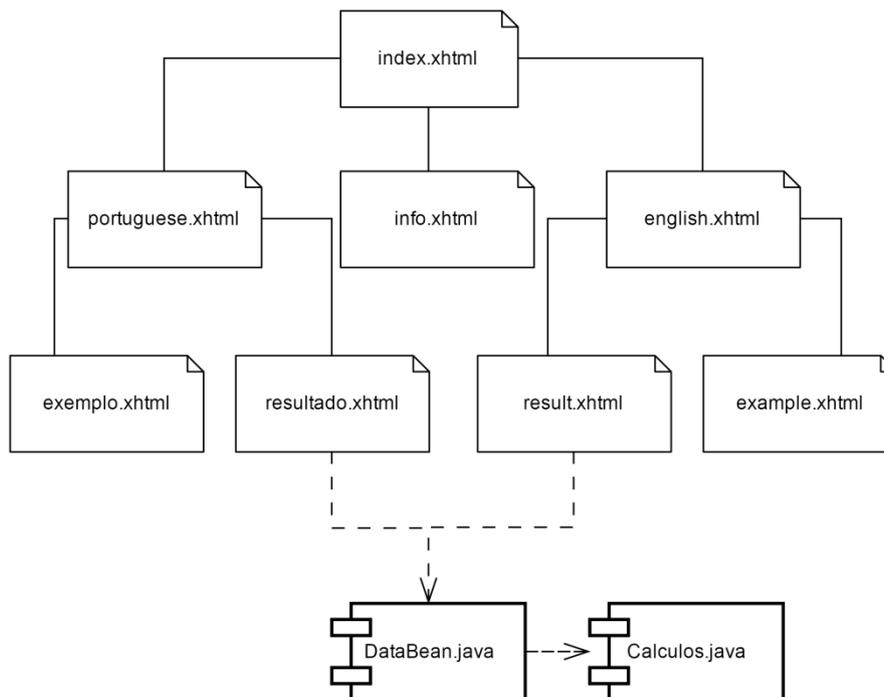


Figura 14 – Diagrama de Páginas

## 3.4. Classes, Métodos Importantes

### 3.4.1. Janela Java Swing

- **Calculos.java**

Esta classe é responsável por todos os cálculos das variáveis em todas as iterações até chegar ao resultado final “Cálculo da quantidade ótima”.

Métodos Importantes:

- *public double processa()*

Este método não recebe nenhum argumento e calcula as variáveis em cada iteração através de um ciclo *do while* onde irá utilizar valores resultantes de outros métodos.

- **JanelaCliente.java**

Esta classe é responsável pela interface Java Swing que vai utilizar informação da classe Calculos.java

Métodos Importantes:

- *public void actionPerformed(ActionEvent e)*

Este método é idêntico ao *processa*, onde é definido quando um determinado botão é pressionado, um determinado método é acionado, criando uma instância desse mesmo método.

## 3.4.2. Interface Web

- **Calculos.java**

Idêntico ao JavaSwing.

- **DataBean.java**

Esta classe é responsável pela comunicação entre a biblioteca de otimização, Calculos, e a aplicação web, nomeadamente os campos nos formulários mostrados em portuguese.xhtml e english.xhtml.

- **index.xhtml**

Esta é a página principal do website.

- **portuguese.xhtml**

Esta é a página onde é apresentado o formulário de preenchimento, no idioma português.

- **english.xhtml**

Esta é a página onde é apresentado o formulário de preenchimento, no idioma inglês.

- **resultado.xhtml**

Esta é a página onde são apresentados todos os resultados dos cálculos efetuados, no idioma português.

- **result.xhtml**

Esta é a página onde são apresentados todos os resultados dos cálculos efetuados, no idioma inglês.

- **exemplo.xhtml**

Esta é a página onde é apresentado um exemplo de preenchimento do formulário, no idioma português.

- **example.xhtml**

Esta é a página onde é apresentado um exemplo de preenchimento do formulário, no idioma inglês.

- **info.xhtml**

Esta é a página onde é apresentada a informação sobre âmbito de desenvolvimento do programa bem como alguns contactos em caso de dúvida, no idioma português e inglês.

## 3.5. Requisitos

Em particular, procedeu-se ao estudo de um sistema de aprovisionamento, tendo-se realizado:

- Seleção do idioma do website em português ou inglês;
- Apresentação de um exemplo de preenchimento dos campos;
- Cálculo da quantidade ótima de encomenda através do preenchimento de um formulário;
- Na página de resultado deverão ser apresentados os valores preenchidos anteriormente; deverá referir qual é o valor da probabilidade de rotura; o valor do nível de encomenda; qual o número médio de unidades por ciclo; qual é a quantidade ótima de encomenda; quais os valores dos cálculos intermédios e por que razão o algoritmo parou, se foi porque o algoritmo atingiu o número máximo de iterações ou se o algoritmo atingiu o limiar de diferença entre a quantidade ótima atual e a quantidade ótima anterior.

- Utilização de um modelo de otimização de stocks estocástico;

Como suporte dos objectivos definidos e respectiva demonstração de resultados foi desenhado um website na plataforma NetBeans IDE 8.0.2, onde a linguagem de programação foi Java.

## 3.6. Hiperligações

### 3.6.1. Aplicação web

A aplicação web está alojada no seguinte endereço web:

<http://alunos.di.uevora.pt:8080/otimizacaodestocks/>

### 3.6.2. Executáveis

A versão executável da aplicação com o interpretador Java (JAR) está alojada no seguinte endereço:

<http://host.di.uevora.pt/~jsaias/optim/>

Nota: Para correr a versão executável tem de ter o JRE (Java Runtime Environment) instalado na máquina.

## Capítulo 4:

# Conclusões e Trabalho Futuro

Este trabalho teve como objetivo desenvolver uma ferramenta que calcule iterativamente a quantidade ótima a encomendar utilizando modelos de otimização global dum sistema de stocks com penúria permitida e com tempo de reposição e procura aleatórias sob suposição de baixas probabilidades de rotura por ciclo ( $\alpha$ ).

Esta ferramenta de apoio à decisão foi desenvolvida com dois layouts, Website e Java Swing (versão portátil que não necessita de acesso à internet).

Utilizando o método de desenvolvimento de Software designado por Prototipação ou simplesmente Protótipo, permitiu desenvolver a partir de uma versão inicial da ferramenta ir refinando até se tornar na versão final. Com a utilização deste método permitiu e facilitou detetar e melhorar falhas e funcionalidades da ferramenta.

A gestão de stocks foi uma área na qual não possuía conhecimento e tornou-se numa mais valia para a formação académica, pois foi necessário estudar o tema e conciliá-lo com a área na qual já possuía conhecimento, a informática.

Uma das limitações desta ferramenta é o facto de o algoritmo utilizado ter aproximação para probabilidades de rotura baixas, mais ou menos, entre 0 a 10%.

Verificou-se também que se podia resolver um problema que exigia iterações fastidiosas e com consultas a duas tabelas (Tabela da normal e da Função de perdas normal) com grande facilidade numa folha de cálculo. Com a facilidade de acesso à Internet existente hoje em dia, decidimos implementar o algoritmo num servidor para possibilitar o livre acesso a esta ferramenta.

Como trabalho futuro seria importante estender a aplicação para suportar outras distribuições matemáticas, criar um módulo que efetuasse uma simulação através de histórico e ainda ultrapassar a limitação do alfa, isto é, a formulação conduzir a probabilidades de rotura superiores à unidade.



# Referências Bibliográficas

BALLOU, Ronald H. (2004), "Business Logistics/Supply Chain Management", 5ª edição, Pearson.

Bergsten H., JavaServer Faces, O'Reilly Media, 2004

Bronson, R.; G. Naadimuthu. (2001) Investigação Operacional. Coleção Schaum (2ª. Ed.). McGraw-Hill Portugal.

CARVALHO, José Crespo (2004), "A Lógica da Logística", 1ª edição, Lisboa, Edições Sílabo.

CARVALHO, José Crespo (2010), "Logística e Gestão da Cadeia de Abastecimento", 1ª edição, Lisboa, Edições Sílabo.

CHOPRA, S. et al (2004), "Supply Chain Management: Strategy, planning, and control", 2ª edição, NJ, Pearson Education Inc.

CHRISTOPHER, Martin (2005), "Logistics and Supply Chain Management, Creating Value-Added Networks".

COSTA, R. (2002), "Elementos de Apoio às Aulas de Introdução à Investigação Operacional", Licenciatura em Matemática, Departamento de Matemática, FCT/UNL.

Geary D., Horstmann G., Core JavaServer Faces (3rd Edition), Prentice Hall, ISBN-10: 0137012896

Hillier, F., Lieberman, G. (2001). Introduction To Operations Research, Seventh Edition, McGraw Hill.

Khare T., Apache Tomcat 7 Essentials, Packt Publishing, 2012

Tavares, L. V., Oliveira, R.C., Themido, Isabel H. E Correia, F.N. (1996). Investigação Operacional. Lisboa: McGraw-Hill.

The Java™ Tutorials, Oracle Java Documentation [Consultada em 18 de Fevereiro de 2015], Disponível em <http://docs.oracle.com/javase/tutorial/>

Wiley Nicholas S., Professional Java for Web Applications, 2014, ISBN: 978-1-118-65646-4



# **Anexos I:**

## **Manual do Utilizador (em português)**





Universidade de Évora

## Manual do Utilizador (em português)



*Fonte: Página da LogexServices*



## Índice

Introdução.....	68
Janela Java Swing .....	69
Instalação da Aplicação .....	69
Java SE Runtime Environment (JRE) mais recente.....	69
Aplicação .....	69
▪ Janela Java Swing .....	69
▪ Campos.....	70
▪ Botões .....	71
▪ Limitações do Programa.....	71
Possíveis Erros .....	71
Exemplo de Demonstração .....	72
Interface Web.....	74
Instalação da Aplicação .....	74
Diagrama de Componentes.....	75
Diagrama de Páginas.....	75
Aplicação .....	76
▪ Página Inicial.....	76
▪ Informação .....	77
▪ Formulário.....	77
▪ Página de Exemplo de Preenchimento .....	78
▪ Página de Resultados .....	79
▪ Campos.....	80
▪ Botões .....	80
Possíveis Erros .....	81
Exemplo de Demonstração .....	82
Notas Finais .....	84
Contactos .....	84



## Introdução

Este documento tem como objetivo dar apoio ao utilizador final da aplicação, demonstrando todas as funcionalidades presentes no Software, exemplificando passo a passo todas as opções para daí resultar um aproveitamento máximo por parte do utilizador.

O documento apresenta uma estrutura bastante simples e bem esquematizada de modo a facilitar a sua consulta.

Inicialmente é apresentada uma breve explicação sobre requisitos e métodos de instalação, de modo a que a aplicação fique pronta a ser utilizada.

Este Software é composto por duas interfaces visuais distintas, o que significa que neste caso irá ser tratado como dois Softwares distintos mas no mesmo manual.

Inicialmente será apresentado o Manual Utilizador para a janela Java Swing e de seguida para a interface web.

# Janela Java Swing

---

## Instalação da Aplicação

### Hardware:

Qualquer processador;  
Disco rígido espaço ocupado 2MB;  
256 MB de memória RAM.

### Software:

Sistema Operativo: Qualquer;  
Java SE Runtime Environment (JRE) mais recente.

**Nota:** O Software não funcionará se não tiver o Java SE Runtime Environment (JRE) instalado na máquina.

## Aplicação

- **Janela Java Swing**

Este Software apenas contém uma janela que é o que é visualizado assim que inicia a aplicação.



*Figura 1 – Menu Inicial*

Através da *Figura 1* podemos ter uma visão do Layout da aplicação, que será explicado de seguida.

- Nesta janela podemos visualizar uma série de campos;
- No canto inferior esquerdo e no canto inferior direito encontra-se dois botões, respectivamente SAIR e CALCULAR.

## ▪ Campos

- Custo de Encomenda [euros]  
Este campo deverá ser preenchido com o valor total, em euros, de cada encomenda.
- Custo de Pose [euros.tonelada<sup>-1</sup>]  
Este campo deverá ser preenchido com o valor total, em euros, de gastos com o armazenamento por unidade de peso, em toneladas, durante um ano.
- Custo de Rotura [euros.tonelada<sup>-1</sup>]  
Este campo deverá ser preenchido com o valor a pagar, em euros, caso haja escassez de material de forma interna ou externa, por unidade de peso, em toneladas, durante um ano.
- Procura Média [toneladas.ano<sup>-1</sup>]  
Este campo deverá ser preenchido com o consumo anual em toneladas.
- Tempo Médio de Reposição [dias]  
Este campo deverá ser preenchido com o tempo médio que decorre entre a colocação da sua encomenda e a sua entrega.
- Variância da Procura Durante o Tempo de Reposição [toneladas<sup>2</sup>]  
Este campo deverá ser preenchido com o valor da variância da procura durante o tempo de reposição, em toneladas<sup>2</sup>.

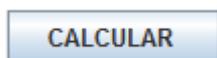
## Validação dos campos

- ✓ Todos os campos devem ser devidamente preenchidos;
- ✓ Os campos devem ser preenchidos somente com números;
- ✓ Caso o valor tenha casas decimais, estas devem ser representadas por “.” (ponto)

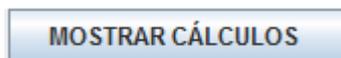
### ▪ **Botões**



Este botão permite finalizar a aplicação



Este botão permite iniciar o cálculo da quantidade ótima baseada nos dos valores preenchidos nos campos.



Este botão permite mostrar os cálculos intermédios dos até chegar ao valor da quantidade ótima .

Nota: Este botão aparece após pressionar o botão CALCULAR.

### ▪ **Limitações do Programa**

O algoritmo utilizado neste projeto tem aproximação para probabilidades de rotura baixas, mais ou menos, entre 0 a 10%.

## **Possíveis Erros**

Em seguida serão apresentados alguns dos problemas que podem surgir:

- Caso não preencha algum campo;
- Caso coloque uma “,” em vez de um “.” como separação de casa decimal;

Todos os erros foram tratados da mesma forma, por isso a mensagem de erro é comum para todos os erros.

**RESULTADO**

Os campos devem ser preenchidos com valores numéricos!

Figura 2 – Exemplo de Preenchimento

## Exemplo de Demonstração

Inicialmente preenche-se os campos com os valores como ilustra a Figura 3.

Implementação Dum Processo Iterativo Para Otimização de Stocks

Cálculo da Quantidade Ótima

Custo da Encomenda [euros]	<input type="text" value="80"/>	Custo de Pose [euros.tonelada <sup>-1</sup> ]	<input type="text" value="16"/>
Custo de Rotura [euros.tonelada <sup>-1</sup> ]	<input type="text" value="150"/>	Procura Média [toneladas.ano <sup>-1</sup> ]	<input type="text" value="44.4"/>
Tempo Médio de Reposição [dias]	<input type="text" value="60"/>	Variância da Procura Durante o Tempo de Reposição [toneladas <sup>2</sup> ]	<input type="text" value="5.92"/>

RESULTADO

SAIR CALCULAR

Figura 3 – Exemplo de Preenchimento Janela

De seguida pressiona-se o botão CALCULAR.

Implementação Dum Processo Iterativo Para Otimização de Stocks

Cálculo da Quantidade Ótima

Custo da Encomenda [euros]	<input type="text" value="80"/>	Custo de Pose [euros.tonelada <sup>-1</sup> ]	<input type="text" value="16"/>
Custo de Rotura [euros.tonelada <sup>-1</sup> ]	<input type="text" value="150"/>	Procura Média [toneladas.ano <sup>-1</sup> ]	<input type="text" value="44.4"/>
Tempo Médio de Reposição [dias]	<input type="text" value="60"/>	Variância da Procura Durante o Tempo de Reposição [toneladas <sup>2</sup> ]	<input type="text" value="5.92"/>

RESULTADO

The optimal amount is: 22,123

SAIR MOSTRAR CÁLCULOS

Figura 4 – Resultado

Na figura acima podemos constatar que no visor do RESULTADO está disposto o resultado, para este exemplo: “A quantidade óptima é: 22.123”. Podemos constatar também que no lugar do botão CALCULAR surgiu agora o botão MOSTRAR CÁLCULOS.

Caso pretenda visualizar os cálculos de todas as iterações até chegar ao resultado mostrado pressione no botão MOSTRAR CÁLCULOS.

Implementação Dum Processo Iterativo Para Otimização de Stocks

Cálculo da Quantidade Ótima

Custo da Encomenda [euros]

Custo de Pose [euros.tonelada<sup>-1</sup>]

Custo de Rotura [euros.tonelada<sup>-1</sup>]

Procura Média [toneladas.ano<sup>-1</sup>]

Tempo Médio de Reposição [dias]

Variância da Procura Durante o Tempo de Reposição [toneladas<sup>2</sup>]

RESULTADO

Iteração 1  
Q 21.071307505705477 ; alfa 0.050621759773466606 ; Z 1.6388547211994016 ; M 11.286135773271923 ; dzeta 0.021194766193150638 ; eta 0.05156909185525125

Iteração 2  
Q 22.06651918562365 ; alfa 0.053012658704200966 ; Z 1.6163192015494448 ; M 11.2313044874609 ; dzeta 0.022362329249635243 ; eta 0.05440989537994926

Iteração 3  
Q 22.120041543898775 ; alfa 0.05314124094630336 ; Z 1.6151302937080203 ; M 11.228411749832983 ; dzeta 0.02242543282700961 ; eta 0.054563433010341486

Iteração 4  
Q 22.122930592060115 ; alfa 0.05314818160254683 ; Z 1.6150661833448612 ; M 11.22825576258705 ; dzeta 0.02242883995374592 ; eta 0.05457172290748053

Iteração 5  
Q 22.123086568570795 ; alfa 0.0531485563208908 ; Z 1.6150627222862182 ; M 11.22824734146792 ; dzeta 0.022429023903367484 ; eta 0.054572170476226735

SAIR

MOSTRAR CÁLCULOS

Figura 5 – Mostrar Cálculos

Após visualizar os valores calculados em cada iteração pode fechar a aplicação pressionando no botão SAIR.

## Interface Web

---

### Instalação da Aplicação

A aplicação usa uma arquitectura cliente servidor.

Na máquina que funcionará como servidor:

**Hardware:**

Qualquer processador;  
Disco rígido espaço ocupado 11MB;  
256 MB de memória RAM;  
Acesso a internet.

**Software:**

Sistema Operativo: Qualquer;  
Servidor: Servidor Aplicaçional Apache Tomcat

Na máquina dos clientes:

**Hardware:**

Qualquer processador;  
Disco rígido espaço ocupado 2MB;  
256 MB de memória RAM;  
Acesso a internet.

**Software:**

Sistema Operativo: Qualquer;  
Navegador de Internet: Qualquer navegador.

## Diagrama de Componentes

Abaixo será apresentado o diagrama de componentes que esquematiza a ligação entre o cliente e o servidor.

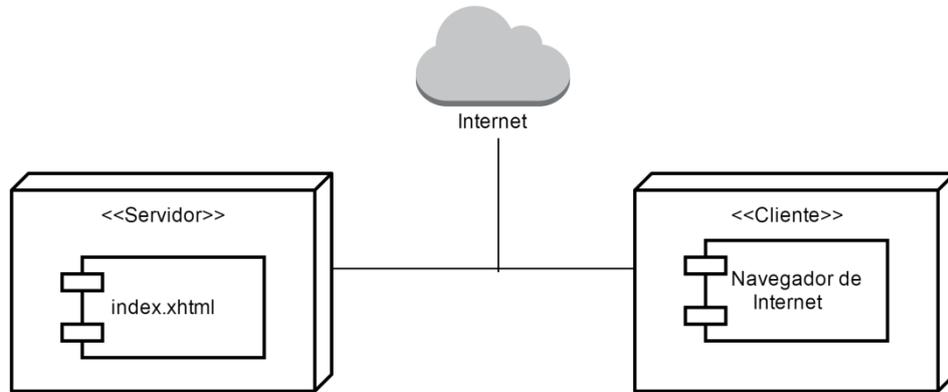


Figura 6 – Diagrama de Componentes

## Diagrama de Páginas

Neste ponto do documento será apresentado um diagrama de páginas que esquematiza toda a estrutura da aplicação.

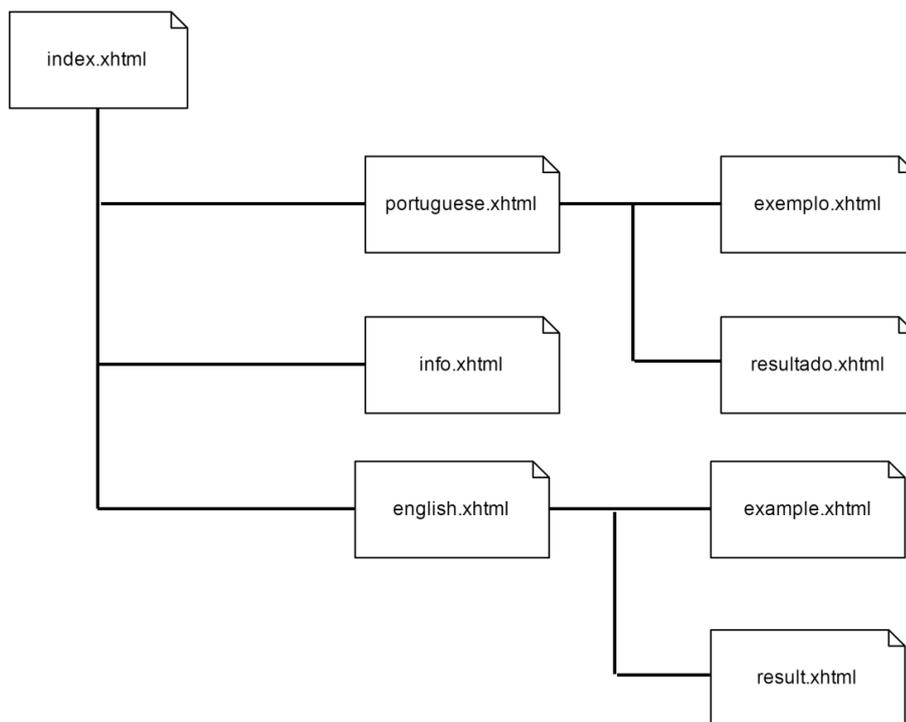


Figura 7 – Diagrama de Páginas

## Aplicação

### ▪ Página Inicial

A página inicial, como o próprio nome indica, é a página que é visualizada assim que inicia a navegação na aplicação. Ao aceder ao endereço da página esta é carregada por defeito.



#### Processo Iterativo para a Otimização de Stocks

De um modo geral esta aplicação consiste em calcular iterativamente a quantidade ótima a encomendar tendo como objetivo a otimização de stocks.

#### Iterative Process for Inventory Optimization

In general, this application consists in calculating iteratively the optimal quantity to order with the objective of inventory optimization.



*Este projeto tem o apoio do Centro de Inovação em Tecnologias da Informação da Universidade de Évora.*

*This project has the support of Innovation Center in Technologies of Universidade de Évora.*

*Figura 8 – Página Inicial*

Através da *Figura 8* podemos escolher o idioma da aplicação, Português ou Inglês.

As funcionalidades permanecem iguais independentemente da escolha do idioma.

A explicação do funcionamento irá incidir sobre o idioma em Português.

Ao clicar no ícone “i” irá para uma página onde será apresentada informação e contactos, como se pode ver na *Figura 9*.

Ao clicar no ícone da bandeira de Portugal irá para uma página como se pode ver na *Figura 10*.

## ▪ Informação

Nesta página pode-se visualizar alguma informação sobre o propósito da aplicação bem como alguns contactos em caso de dúvida.

Esta aplicação foi feita no âmbito da dissertação de mestrado do curso Modelação Estatística e Análise de Dados, pelo aluno Bruno Jorge Pereira da Silva e pelos orientadores, Jorge Manuel Azevedo Santos e José Miguel Gomes Saias, no ano letivo de 2014/2015.

Para mais informações contactar:

*brunojorgesilva84@gmail.com*

*jorgen.santos@gmail.com*

*jsaias@uevora.pt*

This application was developed within Bruno Jorge Pereira da Silva's master degree in Statistical Modelling and Data Analysis course, helped by the tutors Jorge Manuel Azevedo Santos and José Miguel Gomes Saias in the school year of 2014/2015

For more informations contact:

*brunojorgesilva84@gmail.com*

*jorgen.santos@gmail.com*

*jsaias@uevora.pt*

Figure 9 – Informação

## ▪ Formulário

Nesta página pode-se proceder ao preenchimento do formulário com seis campos para proceder ao cálculo da quantidade ótima de encomenda.

### Implementação dum processo iterativo para otimização de stocks

Por favor preencha o formulário abaixo:

Exemplo de Preenchimento

Custo de Encomenda [euros]

0.0

Custo de Pose [euros.tonelada<sup>-1</sup>]

0.0

Custo de Rotura [euros.tonelada<sup>-1</sup>]

0.0

Procura Média [tonelada.ano<sup>-1</sup>]

0.0

Tempo Médio de Reposição [dias]

0.0

Variância da Procura Durante o Tempo de Reposição [tonelada<sup>2</sup>]

0.0

ENVIAR

VOLTAR

Figure 10 – Formulário

Nesta página:

- Encontra-se um botão “Exemplo de Preenchimento”, onde se pode visualizar um exemplo de preenchimento dos dados;
- Pode-se visualizar um formulário para preencher com os valores nos respectivos campos;
- Encontra-se um botão “ENVIAR” que permite submeter os dados e efetuar o cálculo da quantidade ótima;
- E encontra-se um botão “VOLTAR”, que permite voltar à página anterior.

## ▪ Página de Exemplo de Preenchimento

Nesta página é apresentado um exemplo de como preencher os campos.

### Exemplo de preenchimento:

Regra de preenchimento:

- Quando os números tiverem casas decimais, separar por ponto em vez de vírgula.

### Implementação dum processo iterativo para otimização de stocks

Por favor preencha o formulário abaixo:

Exemplo de Preenchimento

Custo de Encomenda [euros]

80.0

Custo de Pose [euros.tonelada<sup>-1</sup>]

16.0

Custo de Rotura [euros.tonelada<sup>-1</sup>]

150.0

Procura Média [tonelada.ano<sup>-1</sup>]

44.4

Tempo Médio de Reposição [dias]

60.0

Variância da Procura Durante o Tempo de Reposição [tonelada<sup>2</sup>]

5.92

VOLTAR

Figura 11 – Exemplo de Preenchimento

## ▪ Página de Resultados

Nesta página pode-se constatar que são apresentadas três informações:

- Dados: são os dados introduzidos no formulário;
- Resultado: contém a quantidade ótima de encomenda, a probabilidade de rotura, o nível de encomenda, o período médio entre encomendas, o número médio de unidades em falta por ciclo e a razão pela qual o programa parou.
- Lista de Iterações, onde são apresentados todos os resultados dos cálculos intermédios até chegar ao resultado da quantidade ótima.

### Dados:

---

Custo de Encomenda[euros]= 0.0      Custo de Pose [euros.tonelada<sup>-1</sup>]= 0.0  
Custo de Rotura [euros.tonelada<sup>-1</sup>]= 0.0 Procura Média [tonelada.ano<sup>-1</sup>]= 0.0  
Tempo Médio de Reposição [dias]= 0.0    Variância da Procura Durante o Tempo de R

### Resultado:

---

A quantidade ótima de encomenda é: 0.  
A probabilidade de rotura é 0.053.  
Sendo o nível de encomenda ótimo de 11.228.  
O período médio entre encomendas é 181.992 dias  
O número médio de unidades em falta por ciclo de 0.055.  
O Programa parou pois convergiu para a solução apresentada. The program stopped

### Lista de Iterações:

---

```
Os valores devem ser superiores a zero!  
The values must be grather than zero!
```

Figura 12 – Página Resultado

## ▪ Campos

São idênticos aos da Janela Java Swing (ver pág.5).

### Validação dos campos

- ✓ Todos os campos devem ser devidamente preenchidos;
- ✓ Os campos devem ser preenchidos somente com números;
- ✓ Caso o valor tenha casas decimais, estas devem ser representadas por “.”
- ✓ O Software foi concebido para aceitar valores de probabilidades de rotura baixas, entre 0 e 10%.

## ▪ Botões



Este botão permite visualizar um exemplo de preenchimento do formulário.



Este botão permite iniciar os cálculos dos valores preenchidos nos campos.

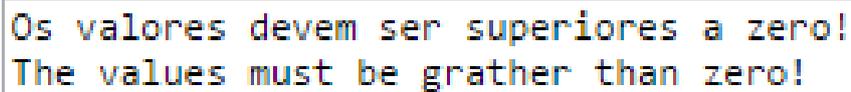


Este botão permite voltar à página anterior.

## Possíveis Erros

Em seguida serão apresentados alguns dos problemas que podem surgir:

- Caso deixe os campos a zero ou algum dos valores seja inferior a 0.



```
Os valores devem ser superiores a zero!  
The values must be grather than zero!
```

Figura 13 – Erro de zero

- Caso deixe algum dos campos completamente vazio.

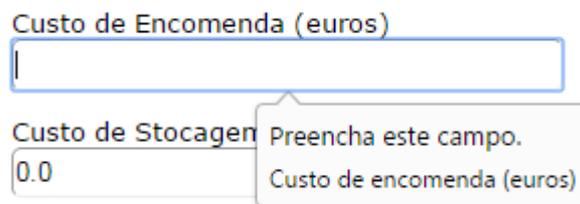


Figura 14 – Erro de não preenchimento

- Caso coloque uma “,” em vez de um “.” como separação de casa decimal;

'80,5' must be a number consisting of one or more digits.

Figura 15 – Erro de separação

## Exemplo de Demonstração

Inicialmente começa-se por preencher os campos com os valores como ilustra a *Figura 16*.



### Processo Iterativo para a Otimização de Stocks

De um modo geral esta aplicação consiste em calcular iterativamente a quantidade ótima a encomendar tendo como objetivo a otimização de stocks.

### Iterative Process for Inventory Optimization

In general, this application consists in calculating iteratively the optimal quantity to order with the objective of inventory optimization.



*Este projeto tem o apoio do Centro de Inovação em Tecnologias da Informação da Universidade de Évora.*

*This project has the support of Innovation Center in Technologies of Universidade de Évora.*

*Figura 16 – Exemplo de Página Inicial*

De seguida preencher-se os campos com os valores como ilustra a *Figura 17*.

### Implementação dum processo iterativo para otimização de stocks

Por favor preencha o formulário abaixo:

Exemplo de Preenchimento

Custo de Encomenda [euros]

80.0

Custo de Pose [euros.tonelada<sup>-1</sup>]

16.0

Custo de Rotura [euros.tonelada<sup>-1</sup>]

150.0

Procura Média [tonelada.ano<sup>-1</sup>]

44.4

Tempo Médio de Reposição [dias]

60.0

Variância da Procura Durante o Tempo de Reposição [tonelada<sup>2</sup>]

5.92

ENVIAR

VOLTAR

*Figura 17 – Exemplo de Preenchimento Página*

De seguida pressiona-se o botão “ENVIAR”.

**Dados:**

Custo de Encomenda [euros] = 80.0      Custo de Pose [euros.tonelada<sup>-1</sup>] = 16.0  
Custo de Rotura [euros.tonelada<sup>-1</sup>] = 150.0      Procura Média [tonelada.ano<sup>-1</sup>] = 44.4  
Tempo Médio de Reposição [dias] = 60.0      Variância da Procura Durante o Tempo de Reposição [tonelada<sup>2</sup>] = 5.92

**Resultado:**

A quantidade ótima de encomenda é: 22,123.  
A probabilidade de rotura é 0.053.  
Sendo o nível de encomenda ótimo de 11.228.  
O período médio entre encomendas é 181.992 dias  
O número médio de unidades em falta por ciclo de 0.055.  
O Programa parou pois convergiu para a solução apresentada. The program stopped because it converged to the presented solution.

**Lista de Iterações:**

```
Iteração 1  
Q 21.071307505705477 ; alfa 0.050621759773466606 ; Z 1.6388547211994016 ; M 11.286135773271923 ; dzeta 0.021194766193150638 ; eta 0.05156909185525125  
Iteração 2  
Q 22.06651918562365 ; alfa 0.053012658704200966 ; Z 1.6163192015494448 ; M 11.2313044874609 ; dzeta 0.022362329249635243 ; eta 0.05440989537994926  
Iteração 3  
Q 22.120041543898775 ; alfa 0.05314124094630336 ; Z 1.6151302937080203 ; M 11.228411749832983 ; dzeta 0.02242543282700961 ; eta 0.054563433010341486  
Iteração 4  
Q 22.122930592060115 ; alfa 0.05314818160254683 ; Z 1.6150661833448612 ; M 11.22825576258705 ; dzeta 0.02242883995374592 ; eta 0.05457172290748053  
Iteração 5  
Q 22.123086568570795 ; alfa 0.0531485563208908 ; Z 1.6150627222862182 ; M 11.22824734146792 ; dzeta 0.022429023903367484 ; eta 0.054572170476226735
```

*Figura 18 – Exemplo Página de Resultado*

Os dados introduzidos na página inicial foram:

- Custo de Encomenda [euros] = 80.0
- Custo de Pose [euros.tonelada<sup>-1</sup>] = 16.0
- Custo de Rotura [euros.tonelada<sup>-1</sup>] = 150.0
- Procura Média [toneladas.ano<sup>-1</sup>] = 44.4
- Tempo Médio de Reposição [dias] = 60.0
- Variância da Procura Durante o Tempo de Reposição [toneladas<sup>2</sup>] = 5.92

O resultado dos cálculos para a quantidade ótima de encomenda foi aproximadamente de 22.123.

Na secção Lista de Iterações foram mostrados os cálculos das variáveis necessárias ao cálculo da quantidade ótima.

## Notas Finais

Com a conclusão da consulta deste documento esperamos que o utilizador consiga não só ter uma visão detalhada de todos os pormenores, como no caso de se deparar com algum problema este o consiga resolver sozinho, contudo isso não inviabiliza que caso o problema persista, este não hesite em contactar o suporte técnico da aplicação.

## Contactos

Bruno Silva



[brunojorgesilva84@gmail.com](mailto:brunojorgesilva84@gmail.com)



963 511 594



# **Anexo II:**

## **User Guide (in english)**





Universidade de Évora

## User Guide (in english)



*Font: LogexServices' Website*



## Índex

Introduction.....	92
Window Java Swing .....	93
Latest version of Java SE Runtime Environment (JRE).....	93
Application .....	93
▪ Window Java Swing.....	93
▪ Fields .....	94
▪ Buttons.....	95
▪ Program Limitations.....	95
Possible errors .....	95
Demonstration Example.....	96
Web Interface.....	98
Software Installation.....	98
Component Diagram.....	99
Page Diagram .....	99
Application .....	100
▪ Home Page.....	100
▪ Info.....	101
▪ Form.....	101
▪ Fill Example Page .....	102
▪ Results Page.....	103
▪ Fields .....	104
▪ Buttons.....	104
Possible Errors.....	105
Demonstration Example.....	106
Final Notes.....	108
Contactos.....	108



## Introduction

This document aims to support the end user of the application, showing all the features present in the Software, illustrating step by step all options for maximum enjoy to the user.

The document has a very simple structure and well schematically to facilitate your query.

Initially, a brief explanation is given about requirements and methods of installation, so that the application is ready to be used.

This software consists of two distinct visual interfaces, which means that in this case will be treated as two separate Software but in the same manual.

Initially will be presented to the User Guide for Java Swing window and then to the Web Interface.

# Window Java Swing

---

## Software Installation

### Hardware:

Any processor;  
Hard Disc required space 2MB;  
256 MB of RAM memory.

### Software:

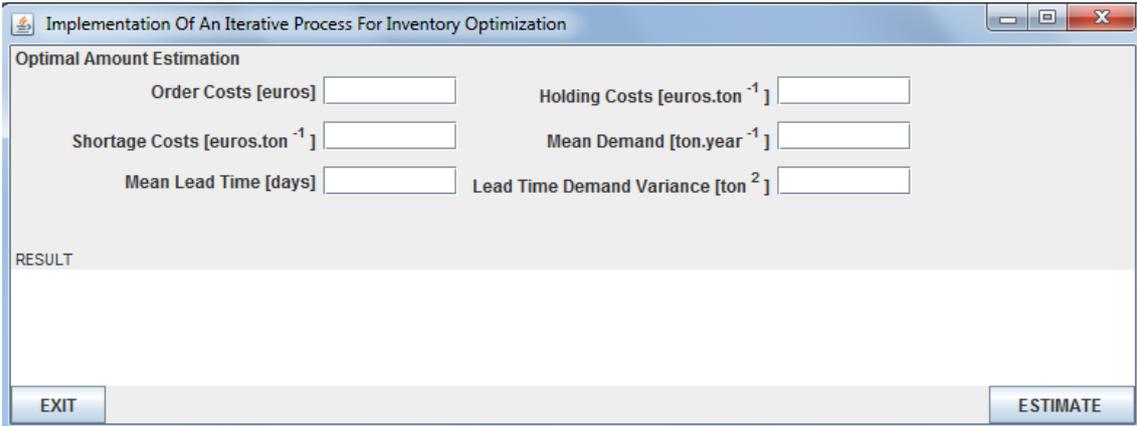
Operating System: Any;  
Latest version of Java SE Runtime Environment (JRE).

**Note:** This software will not work if you do not have the Java SE Runtime Environment (JRE) installed on the machine.

## Application

- **Window Java Swing**

This software only contains a window that is what you see so starts the application.



Implementation Of An Iterative Process For Inventory Optimization

Optimal Amount Estimation

Order Costs [euros]  Holding Costs [euros.ton<sup>-1</sup>]

Shortage Costs [euros.ton<sup>-1</sup>]  Mean Demand [ton.year<sup>-1</sup>]

Mean Lead Time [days]  Lead Time Demand Variance [ton<sup>2</sup>]

RESULT

Figure 1 – Main Menu

In the *Figure 1* we can see the application's layout, which will be explained below.

- In this window you can see some fields;
- In the lower left and lower right course is two buttons respectively EXIT and ESTIMATE.

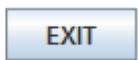
### ▪ **Fields**

- Order Costs [euros]  
This field should be filled with the total payable value, in euros, of each order.
- Holding Costs [euros.ton<sup>-1</sup>]  
This field should be filled with the total payable value, in euros, of spending on storage per unit weight in tons for one year.
- Shortage Cost [euros.ton<sup>-1</sup>]  
This field should be filled with the payable value, in euros, if there is shortage of internal or external material form, per unit weight in tons for one year.
- Mean Demand [ton.year<sup>-1</sup>]  
This field should be filled with the annual consumption in tons.
- Mean Lead Time [days]  
This field should be filled with the mean time elapsing between the placement of your order and its delivery.
- Lead Time Demand Variance [ton<sup>2</sup>]  
This field should be filled with the lead time demand variance, in ton<sup>2</sup>.

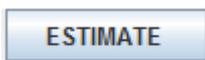
## Fields Validation

- ✓ All fields should be properly filled;
- ✓ The fields should be filled only with numbers;
- ✓ If the value has decimal places, they shall be represented by “.” (period)
- ✓ The algorithm used in this approach to software has low probability of shortage, plus or minus, between 0 and 10%.

### ▪ **Buttons**



This button allows you to end the application.



This button allows you to start the calculation of the optimal amount based on the values filled on the fields.



This button shows the intermediate calculations until reach the value of the optimal amount.

**Note: This button appears after you press the button ESTIMATE.**

### ▪ **Program Limitations**

The algorithm used in this project has approach for low shortage probability, more or less, between 0 to 10%.

## **Possible errors**

Next will be presented some problems may occur:

- If you don't fill any field;
- If you put “,” instead “.” as decimal separation;

Both errors was treated similarly, therefore the error message is common.

**RESULT**  
The fields should be filled with numeric values and grather than zero!

Figure 2 – Fill Example

## Demonstration Example

Initially fill the fields with the values as shown in *Figure 3*.

Implementation Of An Iterative Process For Inventory Optimization

Optimal Amount Estimation

Order Costs [euros]	80	Holding Costs [euros.ton <sup>-1</sup> ]	16
Shortage Costs [euros.ton <sup>-1</sup> ]	150	Mean Demand [ton.year <sup>-1</sup> ]	44.4
Mean Lead Time [days]	60	Lead Time Demand Variance [ton <sup>2</sup> ]	5.92

RESULT

EXIT ESTIMATE

Figure 3 – Fill Example Window

Next you can press in button ESTIMATE.

Implementation Of An Iterative Process For Inventory Optimization

Optimal Amount Estimation

Order Costs [euros]	80	Holding Costs [euros.ton <sup>-1</sup> ]	16
Shortage Costs [euros.ton <sup>-1</sup> ]	150	Mean Demand [ton.year <sup>-1</sup> ]	44.4
Mean Lead Time [days]	60	Lead Time Demand Variance [ton <sup>2</sup> ]	5.92

RESULT

The optimal amount is: 22,123

EXIT SHOW ESTIMATES

Figure 4 – Result

In the figure above we can see on display RESULT is willing the result for this example: “The optimal amount is: 22.123”. We can also see that in place of ESTIMATE button now appeared the SHOW ESTIMATES button.

If you wish to view the calculations of all iterations to reach the result shown press the button SHOW ESTIMATES button.

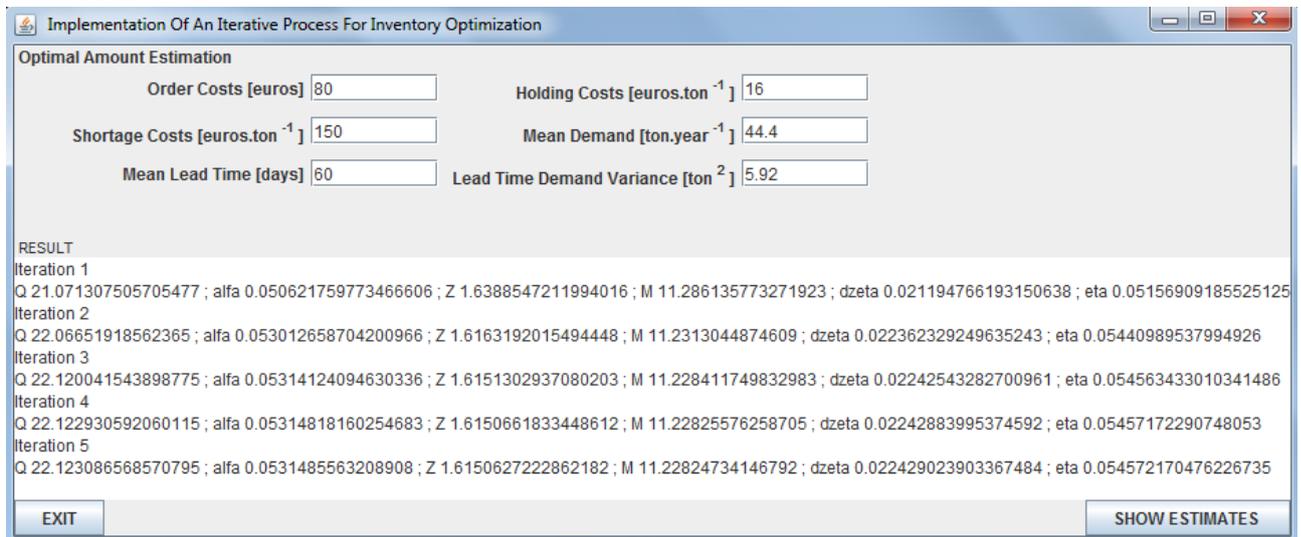


Figure 5 – Show Calculations

After viewing the values calculated in each iteration can close the application by pressing the button EXIT.

# Web Interface

---

## Software Installation

The application has a client-server structure.

### The machine that will act as a server:

#### **Hardware:**

Any processor;  
Hard Disc requirement space 11MB;  
256 MB of RAM memory;  
Internet connection required.

#### **Software:**

Operation System: Any;  
Server: Apache Tomcat Application Server

### The machine that will act as a client:

#### **Hardware:**

Any processor;  
Hard Disc requirement space 2MB;  
256 MB of RAM memory;  
Internet connection required.

#### **Software:**

Operation System: Any;  
Browser: Any browser.

## Component Diagram

Below is will be presented the component diagram that schematizes the connection between the client and the server.

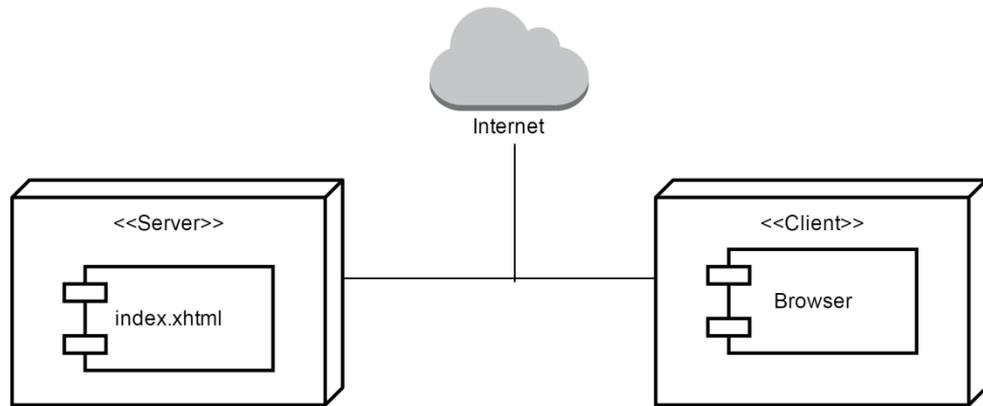


Figure 6 – Component Diagram

## Page Diagram

At this point of the document will be presented a diagram that schematises all entire page's structure of the application.

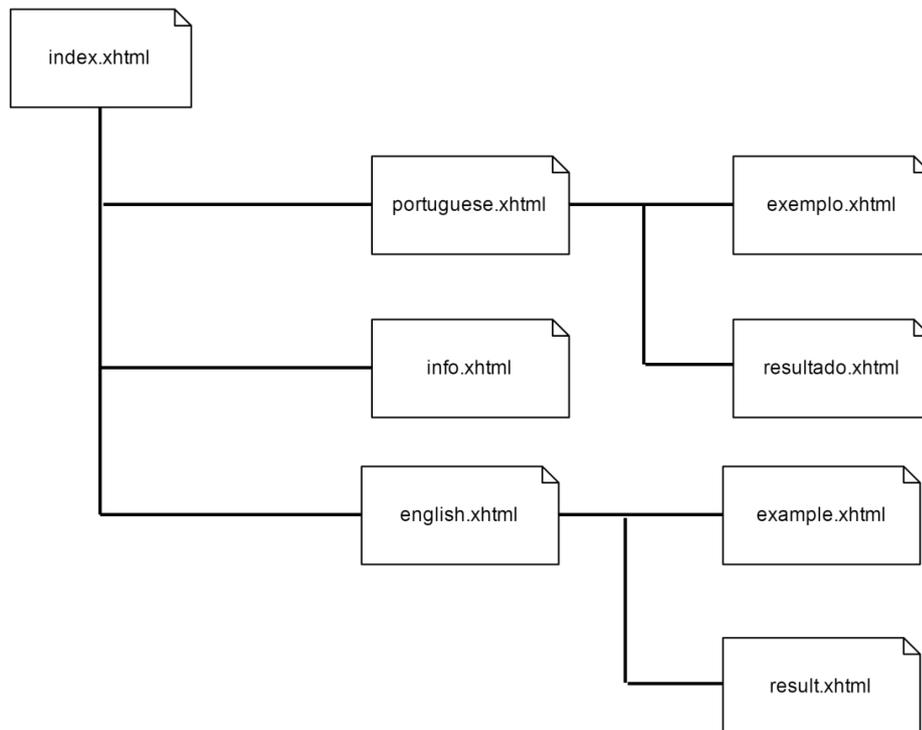


Figure 7 – Page's Diagram

## Application

### ▪ Home Page

The home page, as its name indicates, is the page that is displayed so you start navigating the application. By accessing the page's address this page is loaded by default.



#### Processo Iterativo para a Otimização de Stocks

De um modo geral esta aplicação consiste em calcular iterativamente a quantidade ótima a encomendar tendo como objetivo a otimização de stocks.

#### Iterative Process for Inventory Optimization

In general, this application consists in calculating iteratively the optimal quantity to order with the objective of inventory optimization.



*Este projeto tem o apoio do Centro de Inovação em Tecnologias da Informação da Universidade de Évora.*

*This project has the support of Innovation Center in Technologies of Universidade de Évora.*

*Figure 8 – Home page*

By *Figure 8* we can choose the application language, Portuguese or English.

The features remain the same regardless of the choice of language.

The explanation of the operation will focus on the language in English.

By clicking in “i” icon you will go to the page when you will see some information and contacts, as shown in *Figure 9*.

By clicking in United Kingdom's Flag icon you will go to the page as shown in *Figure 10*.

## ▪ Info

On this page you can see some information about the application's purpose and some contacts in case of doubt.

Esta aplicação foi feita no âmbito da dissertação de mestrado do curso Modelação Estatística e Análise de Dados, pelo aluno Bruno Jorge Pereira da Silva e pelos orientadores, Jorge Manuel Azevedo Santos e José Miguel Gomes Saias, no ano letivo de 2014/2015.

Para mais informações contactar:

*brunojorgesilva84@gmail.com*

*jorgen.santos@gmail.com*

*jsaias@uevora.pt*

This application was developed within Bruno Jorge Pereira da Silva's master degree in Statistical Modelling and Data Analysis course, helped by the tutors Jorge Manuel Azevedo Santos and José Miguel Gomes Saias in the school year of 2014/2015

For more informations contact:

*brunojorgesilva84@gmail.com*

*jorgen.santos@gmail.com*

*jsaias@uevora.pt*

Figure 9 – Info

## ▪ Form

On this page you can proceed to fill the form with six fields in order to calculate the optimal amount to order.

### Implementation of an iterative process for inventory optimization

Please fill the form below:

**Fill Example**

Order Costs [euros]

Holding Costs [euros.ton<sup>-1</sup>]

Shortage Costs [euros.ton<sup>-1</sup>]

Mean Demand [ton.year<sup>-1</sup>]

Mean Lead Time [days]

Lead Time Demand Variance [ton<sup>2</sup>]

**SEND** **BACK**

Figure 10 – Form

On this page:

- We can find a button “Fill Example”, where we can see a fill example with data;
- We can see a form to fill with the respective values in the fields;
- We can find a button “SEND” that allow you to submit the data and perform the calculation of the optimum amount
- We can find a button “BACK”, that allow you to return to the previous page.

## ▪ Fill Example Page

In this page is presented a how to fill example.

### Fill Example:

---

Filling Rule:

- When the numbers have decimal places, separate with point instead of comma.

### Implementation of an iterative process for inventory optimization

---

*Please fill the form below:*

**Fill Example**

Order Costs [euros]	<input type="text" value="80"/>
Holding Costs [euros.ton <sup>-1</sup> ]	<input type="text" value="16"/>
Shortage Costs [euros.ton <sup>-1</sup> ]	<input type="text" value="150"/>
Mean Demand [ton.year <sup>-1</sup> ]	<input type="text" value="44.4"/>
Mean Lead Time [days]	<input type="text" value="60"/>
Lead Time Demand Variance [ton <sup>2</sup> ]	<input type="text" value="5.92"/>

**BACK**

Figure 11 – Fill Example

## ▪ Results Page

On this page you can find that are presented three pieces of information:

- Data: are the data entered into the form;
- Result: contains the optimal amount to order, the shortage probability, the mean period between orders, the mean number of lacking unities and the reason why the program stopped.
- List of iterations, where are presented the results of all intermediate calculations to reach at the result of the optimal amount.

### Data:

---

Order Costs [euros]= 0.0                      Holding Costs [euros.ton<sup>-1</sup>]= 0.0  
Shortage Costs [euros.ton<sup>-1</sup>]= 0.0 Mean Demand [ton.year<sup>-1</sup>]= 0.0  
Mean Lead Time [days]= 0.0              Lead Time Demand Variance [ton<sup>2</sup>]= 0.0

### Result:

---

The optimal amount to order is: 0.  
The shortage probability is 0.053.  
As the optimal order level 11.228.  
The mean period between orders is: 181.992 days  
The mean number of lacking unities per cycle 0.055.  
O Programa parou pois convergiu para a solução apresentada. The program stopped

### Iterations List:

---

```
Os valores devem ser superiores a zero!  
The values must be grather than zero!
```

Figure 12 – Result Page

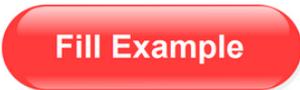
## ▪ **Fields**

Are identical to the Java Swing window (see page 5).

### Fields validation

- ✓ All the fiels must be properly filled;
- ✓ The fields shall be filled only with numbers;
- ✓ If the values have decimal places, they shall be represented by “.”
- ✓ This Software was designed to accept low shortage probability values, between 0 and 10 %.

## ▪ **Buttons**



**Fill Example**

This button allows you to view a filled example form.



**SEND**

This button allows you to start the calculations of the values filled in the fields.



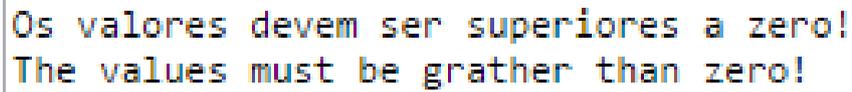
**BACK**

This button allows you to return to the previous page.

## Possible Errors

Next will be presented some of the problems that can arise:

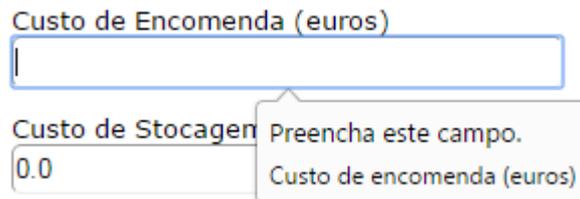
- If you let the fields with zero or some values less than zero.



```
Os valores devem ser superiores a zero!  
The values must be grather than zero!
```

Figure 13 – Zero's Error

- If you let some field empty.



Custo de Encomenda (euros)

Custo de Stocagem 0.0

Preencha este campo.  
Custo de encomenda (euros)

Figure 14 – Filling Error

- If you put a comma “,” instead of a period “.” as decimal separation;

'80,5' must be a number consisting of one or more digits.

Figure 15 – Separation Error

## Demonstration Example

Initially starts up to fill the fields with the values as shown in *Figure 16*.



### Processo Iterativo para a Otimização de Stocks

De um modo geral esta aplicação consiste em calcular iterativamente a quantidade ótima a encomendar tendo como objetivo a otimização de stocks.

### Iterative Process for Inventory Optimization

In general, this application consists in calculating iteratively the optimal quantity to order with the objective of inventory optimization.



*Este projeto tem o apoio do Centro de Inovação em Tecnologias da Informação da Universidade de Évora.*

*This project has the support of Innovation Center in Technologies of Universidade de Évora.*

*Figure 16 – Main Page Example*

Next we have to fill the fields as shown in the *Figure 17*.

### Implementation of an iterative process for inventory optimization

Please fill the form below:

**Fill Example**

Order Costs [euros]

Holding Costs [euros.ton<sup>-1</sup>]

Shortage Costs [euros.ton<sup>-1</sup>]

Mean Demand [ton.year<sup>-1</sup>]

Mean Lead Time [days]

Lead Time Demand Variance [ton<sup>2</sup>]

**SEND** **BACK**

*Figure 17 – Filling Exemple Page*

Next press the button “SEND”.

### Data:

---

Order Costs [euros]= 80.0	Holding Costs [euros.ton <sup>-1</sup> ]= 16.0
Shortage Costs [euros.ton <sup>-1</sup> ]= 150.0	Mean Demand [ton.year <sup>-1</sup> ]= 44.4
Mean Lead Time [days]= 60.0	Lead Time Demand Variance [ton <sup>2</sup> ]= 5.92

### Result:

---

The optimal amount to order is: 22,123.  
The shortage probability is 0.053.  
As the optimal order level 11.228.  
The mean period between orders is: 181.992 days  
The mean number of lacking unities per cycle 0.055.  
O Programa parou pois convergiu para a solução apresentada. The program stopped

### Iterations List:

---

Iteraçã 1
Q 21.071307505705477 ; alfa 0.050621759773466606 ; Z 1.6388547211994016 ; M 11.286:
Iteraçã 2
Q 22.06651918562365 ; alfa 0.053012658704200966 ; Z 1.6163192015494448 ; M 11.2313:
Iteraçã 3
Q 22.120041543898775 ; alfa 0.05314124094630336 ; Z 1.6151302937080203 ; M 11.2284:
Iteraçã 4
Q 22.122930592060115 ; alfa 0.05314818160254683 ; Z 1.6150661833448612 ; M 11.2282:
Iteraçã 5
Q 22.123086568570795 ; alfa 0.0531485563208908 ; Z 1.6150627222862182 ; M 11.22824:

Figure 18 – Result Page Example

The data entered on the home page were:

- Order Costs [euros] = 80.0
- Holding Costs [euros.ton<sup>-1</sup>] = 16.0
- Shortage Cost [euros.ton<sup>-1</sup>] = 150.0
- Mean Demand [ton.year<sup>-1</sup>] = 44.4
- Mean Lead Time [days] = 60.0
- Lead Time Demand Variance [ton<sup>2</sup>] = 5.92

The result of calculations for the optimal amount to order was approximately 22.123.

In the Iteration List section were shown the calculations of the necessary variables of optimal amount.

## Final Notes

With the completion of this consultation document we hope that the user can take a detailed view at all the details, and in the case of encountering any problems this can solve alone, however it does not prevent that if the problem persists, please contact our technical support application.

## Contactos

Bruno Silva



brunojorgesilva84@gmail.com



963 511 594