

**DAVID JORGE CARDOSO PEREIRA**

## **EZ-BUD - ORÇAMENTAÇÃO NA NUVEM**

Dissertação apresentada na Universidade Lusófona de  
Humanidades e Tecnologias para a obtenção de grau de Mestre.

Orientador: Professor Doutor Nuno M. Garcia

**Universidade Lusófona de Humanidades e Tecnologias**  
**ECATI - Escola de Comunicação, Artes e Tecnologias da Informação**

**Lisboa**

**2012**

## **Agradecimentos**

Gostaria de agradecer a todos os que me ajudaram a tornar possível este trabalho, nomeadamente ao Professor Nuno Garcia que, com as suas sábias sugestões o enriqueceu.

Não poderia, claro, esquecer-me do meu filho João David que tem sido, e sempre será, uma motivação constante.

## **Resumo**

Não é novidade que o paradigma vigente baseia-se na Internet, em que cada vez mais aplicações mudam o seu modelo de negócio relativamente a licenciamento e manutenção, para passar a oferecer ao utilizador final uma aplicação mais acessível no que concerne a licenciamento e custos de manutenção, já que as aplicações se encontram distribuídas eliminando os custos de capitais e operacionais inerentes a uma arquitetura centralizada.

Com a disseminação das Interfaces de Programação de Aplicações (Application Programming Interfaces – API) baseadas na Internet, os programadores passaram a poder desenvolver aplicações que utilizam funcionalidades disponibilizadas por terceiros, sem terem que as programar de raiz.

Neste conceito, a API das aplicações Google® permitem a distribuição de aplicações a um mercado muito vasto e a integração com ferramentas de produtividade, sendo uma oportunidade para a difusão de ideias e conceitos.

Este trabalho descreve o processo de conceção e implementação de uma plataforma, usando as tecnologias HTML5, Javascript, PHP e MySQL com integração com ®Google Apps, com o objetivo de permitir ao utilizador a preparação de orçamentos, desde o cálculo de preços de custo compostos, preparação dos preços de venda, elaboração do caderno de encargos e respetivo cronograma.

Palavras-chave: Google apps, WebAPI, Nuvem, PHP, MySQL, JQuery

## **Abstract**

Unsurprisingly the current paradigm is Web based, in which software applications are changing their business model regarding licensing and maintenance, providing the ability to offer end users a more affordable solution regarding licensing and maintenance costs, since the applications are distributed by eliminating the capital and operational costs of a centralized architecture.

With the spread of Application Programming Interfaces (API) based on the Internet, developers were able to develop applications that use features provided by third parties, without having to start from scratch.

In this concept, the API @Google Apps allows deployment of applications to a huge market and integration with productivity tools and an opportunity for the dissemination of ideas and concepts.

This paper describes the process of designing and implementing a platform using technologies like HTML5, Javascript, PHP and MySQL with @Google Apps integration in order to allow the user to the preparation of budgets, starting with the calculation of cost compounds , preparation of sales prices, preparation of the specifications and schedule.

**Keywords:** Google apps, WebAPI, Nuvem, PHP, MySQL, JQuery

## **Abreviaturas, siglas e símbolos**

AAC - Advanced Audio Coding

AJAX - Asynchronous JavaScript and XML

APA - American Psychologic Association

API - Aplicação Programming Interface

CDN - Content Delivery Network

CSS - Cascading Style Sheet

DDL - Data Definition Language ou Linguagem de Definição de Dados

DML - Data Manipulation Language ou Linguagem de Manipulação de Dados

DOM - Document Object Model ou Modelo de objetos de documento

ECMA - European Computer Manufacturers Association

ERP - Enterprise Resource Planning

Gb - Gigabytes

GNU - GNU's Not Unix

GPL - GNU General Public License

GPS - Global Positioning System

GTK - GIMP Toolkit, é uma base multi-plataforma para criar interfaces gráficos

HDD - Hard Disk Drive

HTML - HyperText Markup Language

HTML5 - HyperText Markup Language Versão 5

HTTP - Hypertext Transfer Protocol ou Protocolo de Transferência de Hipertexto

HTTPS - HTTP sobre SSL

IDE - Integrated Development Environment ou Ambiente de Desenvolvimento Integrado

IdP - Identity Provider ou Provedor de Identidade

IMAP - Internet message access protocol

IP - Internet Protocol ou protocolo de comunicação de dados

JIT - Interpretação dinâmica de código, sem pré-compilação.

JSON - JavaScript Object Notation

MKV - Formato multimédia Matroska

MOV - Formato multimédia utilizado pelo software (r)QuickTime

MP4 - Formato multimédia ISO/IEC 14496-14:2003 ou MPEG-4 Part 14

PERL - Linguagem de programação genérica, de alto nível

PHP - Linguagem de programação

POP - Post Office Protocol

RAM - Random Access Memory

RDP Remote Desktop Protocol

REST - Representational State Transfer

RIA - Rich Internet Application

RSS - Really Simple Syndication

SaaS - Software as a Service

SAML - Security Assertion Markup Language

SGBDR ou Sistema de Gestão de Base de Dados Relacional

SOAP - simple Object Access Protocol

SSL - Secure Sockets Layer

SSO - Single Sign-On

TCP - Transmission Control Protocol ou protocolo de controlo de transmissão

UCS - Universal Character Set

UTC - Universal Time Coordinated ou Tempo Universal Coordenado

UTF8 - UCS Transformation Format - 8-bit

Vorbis - Formato Audio de utilização gratuita

W3C - World Wide Web Consortium

WebGL - Web-based Graphics Library ou biblioteca gráfica baseada na internet

XML - Extensible Markup Language

## Índice Geral

1 - Introdução.....	11
2 - Estado da Arte.....	13
2.1 - Evolução do mercado.....	13
2.2 - Nuvem.....	14
2.2.1 - Visibilidade.....	16
2.3 - Requisitos.....	17
2.3.1 - Requisitos Funcionais.....	17
2.3.2 - Requisitos Não Funcionais.....	17
2.4 - Recursos Analisados.....	17
2.4.1 - Primavera Executive.....	18
2.4.2 - Candy/CCS.....	18
2.4.3 - ORSE.....	18
2.4.4 - CORECON.....	19
2.5 - Comparação das soluções.....	19
2.5.1 - Sistema de Autenticação.....	20
2.5.2 - Infraestrutura.....	20
2.5.3 - Agilidade.....	20
2.5.4 - Independência.....	20
2.5.5 - Utilização em Portugal.....	21
2.6 - Importância das APIs .....	21
2.6.1 - Web API (Web Service).....	21
2.6.2 - O aumento de popularidade das WebAPIs.....	22
2.6.3 - WebAPIs mais populares.....	22
2.6.4 - Protocolos Web API.....	22
2.6.5 - Formatos de dados Web API.....	22
2.6.6 - Benefícios das WebAPIs.....	23
2.7 - Importância das Bibliotecas.....	24
2.8 - Tecnologia Adotada.....	24
2.8.1 - W3C.....	25
2.8.2 - HTML.....	25
2.8.3 - CSS.....	26
2.8.4 - HTML5.....	29

2.8.4.1 - Novas funcionalidades do HTML5.....	29
2.8.4.2 - Interfaces de programação.....	33
2.8.5 - JAVASCRIPT.....	37
2.8.6 - PHP.....	38
2.8.7 - MYSQL.....	40
2.8.8 - Formato MPX.....	42
2.9 - APIs utilizadas.....	43
2.9.1 - @Google Apps.....	43
2.9.1.1 - Gmail.....	43
2.9.1.2 - Talk.....	43
2.9.1.3 - Calendar.....	44
2.9.1.4 - Docs.....	44
2.9.1.5 - Sites.....	44
2.9.1.6 - Video.....	44
2.9.1.7 - APIs.....	44
2.9.1.8 - Single Sign On.....	44
2.9.1.9 - Porquê @Google Apps?.....	44
2.9.2 - OpenID.....	45
2.9.3 - Zend.....	50
2.9.4 - JQuery.....	50
3 - A Solução Ez-BUD.....	57
3.1 - 'Stakeholders'.....	57
3.2 - Levantamento de requisitos.....	58
3.2.1 - Requisitos Não funcionais.....	58
3.2.1.1 - Licenciamento.....	58
3.2.1.2 - Requisitos de Implementação.....	58
3.2.1.3 - Requisitos de Fiabilidade.....	59
3.2.1.4 - Requisitos de Segurança.....	59
3.3 - Requisitos Funcionais.....	60
3.3.1 - Preços compostos.....	60
3.3.2 - Orçamentos.....	60
3.3.3 - Cronogramas.....	61
3.4 - Arquitetura da solução.....	62



3.4.1 - Estudo da interface.....	63
3.4.2 - Multi-idioma.....	71
3.4.3 - Arquitetura de dados.....	72
3.4.3.1 - Entidades principais.....	74
3.4.4 - Arquitetura lógica.....	75
3.4.4.1 - Comunicação e mensagens.....	76
4 - Metodologia.....	79
4.1 - Planeamento .....	79
4.2 - Análise de riscos .....	79
4.3 - Fases do projeto.....	80
4.3.1 - Requisitos, Planeamento e Protótipo.....	80
4.3.2 - Desenho e Construção das Bases de Dados.....	80
4.3.3 - Desenvolvimento e Programação.....	80
4.3.4 - Testes à aplicação.....	80
5 - Ferramentas utilizadas.....	81
5.1 - WAMP.....	81
5.2 - Axure RP.....	82
5.3 - Netbeans.....	83
5.4 - PHPDOC.....	84
5.5 - HeidiSql.....	84
5.6 - LibreOffice.....	84
5.7 - GanttProject.....	85
5.8 - Mozilla Firefox.....	85
5.9 - Firebug.....	85
6 - Testes.....	87
6.1 - Implementação.....	87
6.2 - Carga.....	87
6.2.1 - Dados de acesso.....	87
6.2.2 - Tempos de ligação (ms).....	88
6.2.3 - Percentagem dos pedidos servidos por período de tempo(ms).....	88
7 - Conclusão.....	89
8 - Bibliografia.....	90

## Índice de figuras

Figura 1: Previsão de quota de mercado 2008-2015 (Form Factor).....	14
Figura 2: Linguagens de programação utilizadas nos 10000 sites mais visitados, por Builtwith.....	38
Figura 3: Caixa para adicionar a aplicação a um domínio ®Google Apps.....	45
Figura 4: Arquitetura 'Three-tier'.....	62
Figura 5: Ecrã principal do módulo de orçamentos.....	66
Figura 6: Ecrã de edição de orçamentos.....	67
Figura 7: Edição da estrutura do orçamento.....	67
Figura 8: Edição de um item de um orçamento.....	68
Figura 9: Edição de mapas de trabalho.....	69
Figura 10: Edição de clientes.....	69
Figura 11: Edição de preços - separador de preços compostos.....	70
Figura 12: Modelo EER - Entity Relationship Model ou Modelo de Relacionamentos de Entidades.....	73
Figura 13: Fluxo de mensagens - Generalidade.....	76
Figura 14: Fluxo de mensagens – ‘Use Case’ lista de orçamentos.....	77
Figura 15: Diagrama lógico.....	78
Figura 16: Plano de trabalhos (semanas).....	79
Figura 17: Menu de controlo do WampServer.....	81
Figura 18: Ecrã da aplicação Axure RP.....	82
Figura 19: Ecrã da aplicação Netbeans.....	83
Figura 20: Ecrã da aplicação HeidiSQL.....	84
Figura 21: Firebug em depuração.....	86

## Índice de tabelas

Tabela 1: Comparação de soluções.....	19
Tabela 2: Importância das bibliotecas.....	24
Tabela 3: Compatibilidade de selectors CSS e principais navegadores.....	28
Tabela 4: Disponibilidade de serviço.....	59
Tabela 5: Testes - tempos de ligação.....	88

## 1 - Introdução

Um orçamento, tem como objetivo a estimação do preço pelo qual uma entidade, por exemplo uma empresa, se propõe realizar uma dada empreitada, e na perspetiva da entidade que prepara o orçamento é um instrumento extremamente importante na medida em que condiciona fortemente a angariação de obras e consequentemente os resultados financeiros dessa entidade.

Na determinação do preço, este não poderá ser demasiado baixo de forma a não provocar prejuízo à empresa, nem demasiado alto para não levar à perda do respetivo concurso. O orçamento deve, por isso, espelhar o cálculo do somatório dos custos que as empresas prevêem ter com determinada obra, acrescido do óbvio lucro que é o objetivo principal de cada empresa, lucro esse que pode ser calculado de muitas maneiras, incluindo o estimado pelo valor de mercado. Ocorre, no entanto frequentemente, a situação em que uma empresa elabora os seus orçamentos de forma incorreta, ou porque se limita a aplicar os preços de venda do mercado, não conhecendo os seus custos e margens de lucros, ou através do cálculo dos custos por defeito, considerando apenas os mais evidentes e esquecendo outros como custos de equipamentos e custos indiretos levando a propostas muito baixas ou a previsões de lucros que mais tarde se revelam falsas.

É para minimizar este tipo de erros que o software de orçamentação existe desde os primórdios da informática, com os mais variados formatos, facilitando o trabalho aos orçamentistas no cálculo exato de todas as variáveis envolvidas a uma obra. Não obstante existir software que preencha todas as necessidades dos orçamentistas e responsáveis, este teima em não acompanhar o paradigma atual que visa essencialmente a mobilidade e a diminuição de custos.

O facto é que, neste momento, nas empresas distribuidoras de software de orçamentação e respetivos serviços, existe uma preocupação constante em controlar o custos das obras mas não em reduzir, ou mesmo eliminar, os custos de licenciamento por razões óbvias, assim como os respetivos custos de manutenção e de recursos que têm que estar afetos aos departamentos de orçamentação.

Não é novidade que o paradigma vigente baseia-se na Internet, em que cada vez mais aplicações mudam o seu objeto de negócio, diga-se licenciamento e manutenção, para passar a oferecer ao utilizador final uma aplicação virtualmente gratuita no que concerne a custos de licenciamento e extremamente vantajosa no que diz respeito a custos de

manutenção uma vez que as aplicações se encontram na 'Nuvem' eliminando custos de equipamentos centralizadores e respetivo pessoal de IT. Em suma:

- Redução do custo de propriedade;
- Redução do custo de manutenção, já que deixa de ser necessário contratar serviços de manutenção;
- Redução do custo com equipamento, já que o cliente não precisa de adquirir servidores, licenças, software de anti-vírus, ou infraestruturas de rede e de segurança informática e de dados.

Relativamente à estrutura desta dissertação, que segue a norma APA, começarei por analisar, no Capítulo 2, o Estado da Arte, comparando algumas soluções do mercado, evolução do mercado e considerações do novo paradigma de desenvolvimento. A análise à solução a adotar, seguir-se-á no Capítulo 3, descrevendo o levantamento de requisitos funcionais e não funcionais, a arquitetura da aplicação, tanto no que respeita ao estudo da interface e tecnologias adotadas no desenvolvimento da solução, como à própria estrutura e organização do código e arquitetura de dados desenvolvida.

No Capítulo 4, é descrito o respetivo cronograma assim como a metodologia de desenvolvimento utilizada nesta dissertação, e antecederá a descrição das ferramentas de desenvolvimento adotadas, já no Capítulo 5.

Seguidamente, o Capítulo 6, é reservado à execução dos testes à aplicação desenvolvida e considerações sobre a sua usabilidade e performance. Finalmente, o Capítulo 7 é dedicado à conclusão de todo o trabalho desenvolvido, seguindo-se e respetiva bibliografia.

## **2 - Estado da Arte**

A investigação do Estado da Arte é uma das partes mais importantes de qualquer trabalho, uma vez que faz referência ao que já se tem feito sobre o assunto pesquisado, auxiliando na melhoria e desenvolvimento de novos conceitos e paradigmas.

Como anteriormente descrito, software para orçamentação é um tema que, a detalhar minuciosamente, seria o suficiente para a elaboração de vários volumes de análise que estão naturalmente fora do âmbito deste trabalho.

Esta análise centra-se, acima de tudo, na verificação da necessidade de adotar uma solução já existente no mercado ou se, eventualmente, será necessário o desenvolvimento de uma aplicação que preencha todos os problemas anteriormente mencionados.

Este tipo de software (orçamentação) é dos mais "antigos" no que diz respeito a implementações informáticas e está fortemente ligado a plataformas ERP empresariais que não têm, de todo, a agilidade que se exige hoje em dia.

### **2.1 - Evolução do mercado**

As vendas dos dispositivos móveis e adoção de plataformas Web é prova do novo paradigma de mobilidade.

O estudo da Forrester Research (Epps, Forrester Research, 2010), demonstrado na Figura 1, espelha a adoção de dispositivos móveis versus os convencionais. Neste momento, a venda de equipamento de 'desktop' já é de apenas ~27%, ao passo que os dispositivos que permitem mobilidade já têm uma quota de ~74% e a tendência é clara.

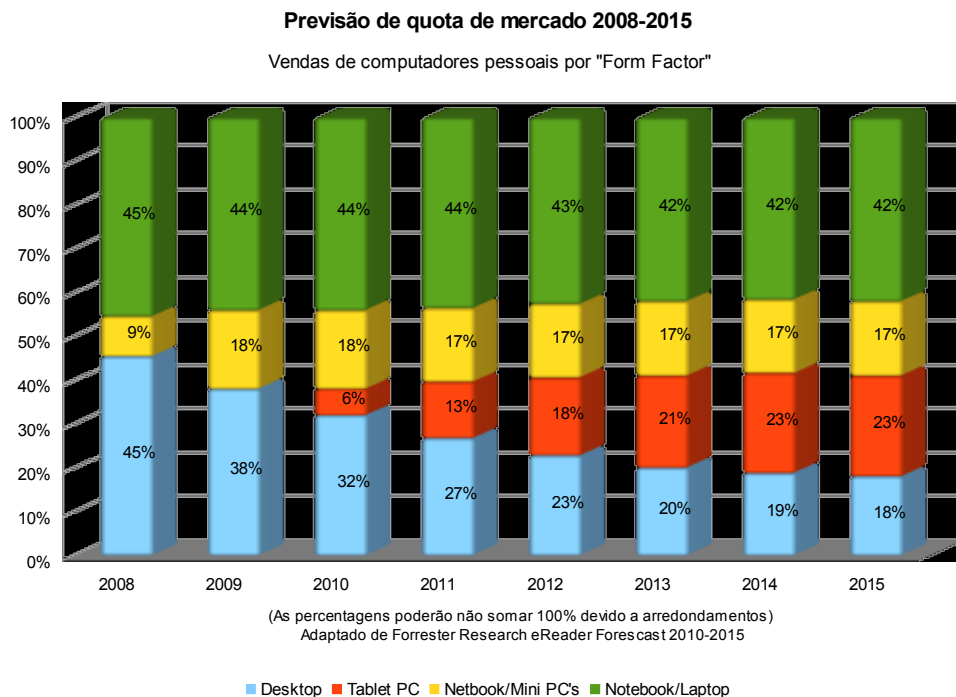


Figura 1: Previsão de quota de mercado 2008-2015 (Form Factor)

Podemos ver mesmo noutras áreas, nomeadamente na Comunicação Social, grandes movimentações no sentido de não perder espaço nesta luta pela mobilidade, através de 'spin-offs' e/ou reestruturações para disponibilizarem edições específicas para este tipo de dispositivos.

Assim podemos concluir que as soluções de software deverão seguir esta tendência sob risco de por em perigo a sua própria sobrevivência.

## 2.2 - Nuvem

O conceito de computação em nuvem ou 'Cloud computing' refere-se à utilização da memória e das capacidades de armazenamento e processamento de computadores e servidores interligados por meio da Internet, seguindo o princípio da computação em grelha.

O armazenamento de dados é feito em serviços que poderão ser acedidos de qualquer parte do mundo, ininterruptamente, não havendo necessidade de instalação de programas ou de armazenar dados localmente. O acesso a programas, serviços e arquivos é remoto, através da Internet, sendo que, o uso deste modelo é mais fiável do que o uso de

unidades físicas.

Assim é possível, a partir de qualquer dispositivo ter acesso a informações, ficheiros e programas, independentemente de plataforma. O requisito mínimo é um computador compatível com os recursos disponíveis na Internet, com os respetivos dispositivos de entrada, teclado e rato, e saída (monitor).

Grandes empresas têm vindo a iniciar uma grande incursão nessa ‘nuvem de informação’, ‘information cloud’, que especialistas consideram uma nova fronteira da era digital dando início a um novo paradigma.

Atualmente, o ‘Cloud Computing’ é dividida em seis tipos:

- IaaS - ‘Infrastructure as a Service’ ou Infraestrutura como Serviço; quando se utilizam recursos de uma infraestrutura, à medida do utilizador.
- PaaS - ‘Platform as a Service’ ou Plataforma como Serviço; utilizando-se plataformas como uma base de dados, um ‘web-service’ ou outro qualquer serviço solicitado pelo cliente, isto é, qualquer utilizador devidamente autorizado.
- DaaS - ‘Development as a Service’ ou Desenvolvimento como Serviço; as ferramentas de desenvolvimento atuam na nuvem como ferramentas partilhadas, ferramentas de desenvolvimento baseadas na web e serviços baseados em partilha de informação.
- SaaS - ‘Software as a Service’ ou Software como Serviço; uso de um software em regime de utilização web, são os casos dos conhecidos ® Google Docs , ® Microsoft Sharepoint Online, etc.
- CaaS - ‘Communication as a Service’ ou Comunicação como Serviço; uso de uma solução de Comunicação Unificada hospedada em Centro de dados do provedor ou fabricante.
- EaaS - ‘Everything as a Service’ ou Tudo como Serviço; quando se utiliza tudo, infraestrutura, plataformas, software, suporte, enfim, o que envolve T.I.C. (Tecnologia da Informação e Comunicação) como um Serviço.



### 2.2.1 - Visibilidade

No que respeita à implementação, depende-se, obviamente, das necessidades das aplicações que serão desenvolvidas. A restrição ou abertura de acesso depende do processo de negócios, do tipo de informação e do nível de visão desejado. Os tipos de implementação quanto à sua visibilidade, são os seguintes:

- Privado - As nuvens privadas são aquelas construídas exclusivamente para um único utilizador, uma empresa, por exemplo. Não sendo um Centro de Dados privado, a infraestrutura utilizada pertence ao utilizador, e, por isso mesmo, este possui total controlo sobre as aplicações e como estas são implementadas.
- Público - As nuvens públicas são aquelas que são geridas por terceiros. As aplicações de diversos utilizadores ficam nos mesmos sistemas de armazenamento, o que pode parecer ineficiente a princípio, porém, se a implementação de uma nuvem pública considera questões fundamentais, como desempenho e segurança, a existência de outras aplicações sendo executadas na mesma nuvem é transparente tanto para os prestadores de serviços como para os utilizadores.
- Comunidade - A infraestrutura de nuvem é partilhada por diversas organizações e suporta uma comunidade específica. Pode ser administrado por organizações ou por um terceiro e pode existir localmente ou remotamente.
- Híbrido - Nas nuvens híbridas temos uma composição dos modelos de nuvens públicas e privadas. Estas permitem que uma nuvem privada possa ter seus recursos ampliados a partir de uma reserva de recursos em uma nuvem pública, característica esta que possui a vantagem de manter os níveis de serviço mesmo que haja flutuações rápidas na necessidade dos recursos. A conexão entre as nuvens pública e privada pode ser usada até mesmo em tarefas periódicas que são mais facilmente implementadas nas nuvens públicas, por exemplo. .

Este novo paradigma tem como principal vantagem a otimização da utilização de recursos assim como a diminuição de custos de manutenção, assim como a abstração da localização da informação ou aplicações fazendo com que os serviços suportados na

Nuvem sejam mais tolerantes a falhas, pelo menos do ponto de vista do utilizador que necessita aceder às suas informações a qualquer hora e em qualquer lugar.

## **2.3 - Requisitos**

Um das dificuldades na análise do estado da arte de determinado tipo de software deve-se à grande diferenciação entre os mesmos no que diz respeito ao detalhe. Para minimizar esta dificuldade teremos em conta apenas alguns requisitos como análise, deixando de lado futuras implementações como a continuação do processo pós-adjudicação.

### **2.3.1 - Requisitos Funcionais**

- Cálculo de Preços e Fichas de Rendimento
- Elaboração de Cadernos de Encargos e Elaboração de Orçamentos
- Cronogramas.

### **2.3.2 - Requisitos Não Funcionais**

- Implementação
  - Aplicação baseada na Internet
  - Portabilidade – Independência do Sistema Operativo.
  - Mobilidade – Conformidade com dispositivos móveis de última geração.
- Processo
  - Conformidade HTML5.
  - Integração Google Apps
  - Licenciamento
- Fiabilidade
- Segurança

## **2.4 - Recursos Analisados**

As soluções analisadas não tiveram em conta quotas de mercado por inexistência de estudos que garantam a veracidade desse tipo de dados uma vez que cada empresa tem os seus números que duvidosamente são reais.

As aplicações atualmente no mercado são imensas pelo que o critério de seleção baseou-se no tipo de distribuição; 'Desktop' com e sem licenciamento; Web e SaaS.

Assim, as aplicações selecionadas foram as seguintes:

1. Primavera Executive ('Desktop' e 'SaaS') (<http://tiny.cc/ult2j> )
2. Candy/CCS (<http://www.ccssa.com/> )
3. ORSE (<http://www.cehop.se.gov.br/orse/> )
4. Corecon (<http://www.corecon.com> )

#### **2.4.1 - Primavera Executive**

Esta solução 'Desktop' ou 'SaaS'<sup>1</sup> é desenvolvido pela 'Primavera Business Software Solutions' e está na versão 7.

É um ERP muito completo, cuja maioria das funcionalidades está fora do âmbito desta dissertação, mas pode também ser utilizado de forma independente compreendo os requisitos funcionais anteriormente descritos; preços, orçamentos e cronogramas.

O modo de licenciamento é por posto de trabalho e por funcionalidade subscrita.

#### **2.4.2 - Candy/CCS**

É uma solução 'Desktop' muito usada em Portugal, desenvolvida pela empresa sul-africana 'Construction Computer Software' e está na versão 2.00a12. O modo de licenciamento é por posto de trabalho, implicando o seu aluguer mensal.

É também possível descarregar o software para avaliação.

#### **2.4.3 - ORSE**

O Software ORSE - Orçamento de Obras de Sergipe, foi desenvolvido e é mantido pela Companhia Estadual de Habitação e Obras Públicas de Sergipe, Brasil - CEHOP há mais de dez anos, para atender à determinação contida nos artigos 8º e 9º da Lei Estadual nº 4.189 de 28.12.1999 que criou o Sistema Estadual de Registro de Preços para Obras e Serviços de Engenharia - Brasil.

É uma aplicação de ambiente de trabalho tradicional e é aparentemente muito usado e completo devido à partilha de informação entre os seus utilizadores sendo que, atualmente a base de dados conta com 8468 itens e 8549 composições de preços unitários.

O sistema é disponibilizado de forma gratuita, propiciando o acesso fácil e rápido das informações a toda a comunidade técnica, empresarial, científica e órgãos de fiscalização e controle.

---

1 A versão SaaS é apenas o alojamento da aplicação num Centro de Dados com acesso via RDP.

## 2.4.4 - CORECON

Este software foi desenvolvido pela Corecon Technologies, Inc. É uma solução baseada na Internet e o seu modo de licenciamento é por utilizador com um custo de \$USD 60 por ano ao que acrescem custos caso a informação supere 1Gb de informação, tendo vários escalões de pagamento com aproximadamente \$USD 5 por Gb/Ano.

## 2.5 - Comparação das soluções

Através da análise da Tabela 1, poderemos inferir que, de facto, não obstante serem todas eficazes no que diz respeito aos requisitos seleccionados, a esmagadora maioria baseia-se em tecnologia, dita tradicional, de ambiente de trabalho, extremamente dependentes de Sistemas Operativos e Apoio Técnico.

Requisitos	PRIMAVERA CONSTRUCTION		Candy/CSS	ORSE	Corecon
	Desktop	SaaS			
Requisitos Não funcionais					
Requisitos de Implementação					
Aplicação baseada na Internet (https)	N	N	N	N	S
Portabilidade (Independência do Sistema Operativo)					
Cliente	Win	Win	Win	Win	*
Servidor	Win	Win	Win	Win	Win
Requisitos do processo					
Conformidade HTML5	N/A	N/A	N/A	N/A	N
Integração Google Apps	N/A	N/A	N/A	N/A	N
Requer Licenciamento	S	S	N	N	S
Requisitos de Fiabilidade					
O Serviço deverá ter uma disponibilidade de 99,9%	N/A	99,5	N/A	N/A	S
Requisitos de Segurança					
Comunicações encriptadas SSL-256bits	N/A	N	N/A	N/A	N
Requisitos Funcionais					
Cálculo de Preços e Fichas de Rendimento	S	S	S	S	S
Elaboração de Cadernos de Encargos e Elaboração de Orçam	S	S	S	S	S
Cronogramas financeiros e Mapas de Trabalho	S	S	S	S	S

### Legenda:

N/A – Não Aplicável

Win – Sistema Operativo Windows

Lin – Sistema Operativo Linux

Mac – Sistema Operativo Apple OSX

Sol - Oracle Solaris 11

\* - Qualquer Sistema Operativo com navegador de versão 5 ou superior

*Tabela 1: Comparação de soluções*

Quanto à eficiência das soluções esta não poderá ser analisada sem ter em conta os Requisitos Não Funcionais pelo que apenas a solução 'Corecon' poderia ser candidata ao título de 'eficiente'.

### **2.5.1 - Sistema de Autenticação**

Relativamente a sistema de autenticação todas as soluções baseiam-se em sistemas proprietários não favorecendo a integração com outros sistemas o que é uma desvantagem, pois hoje em dia, as plataformas devem permitir que outras soluções se possam autenticar ou que se possam autenticar noutras plataformas.

### **2.5.2 - Infraestrutura**

Relativamente à Infraestrutura, excetuando a plataforma 'Corecon' e 'Primavera SaaS', as restantes exigem que o utilizador tenha uma infraestrutura com todos os inconvenientes que, hoje em dia, para uma Pequena ou Média Empresa isso acarreta. Custos Elevados e Pessoal Especializado tais como custos obrigatórios para a existência dessa infraestrutura, respetivo custo do equipamento e *upgrades* futuros.

### **2.5.3 - Agilidade**

Garantindo que os nossos dados estão na 'Nuvem' podemos trabalhar com qualquer computador, a qualquer hora do dia e em qualquer local. Temos, por um lado, a segurança dos dados num Centro de Dados especializado, com condições que poucas empresas poderão ter, e por outro, a agilidade de podermos estar onde e quando quisermos ou necessitarmos. Através da utilização dum certificado SSL as comunicações poderão ser cifradas sendo praticamente impossível decifrar os dados trocados de e para o servidor.

Das plataformas analisadas apenas duas contemplam esta questão; 'Corecon' e 'Primavera SaaS', sendo que a 'Primavera SaaS' não dispõe da agilidade que uma aplicação Web oferece, isto é, se uma aplicação Web apenas necessita de um navegador de versão 5 ou superior, esta solução 'SaaS' baseia-se num acesso RDP a uma aplicação alojada num centro de dados, com as desvantagens de ser mais exigente a nível de largura de banda e conforto do próprio utilizador uma vez que trabalhar várias horas por 'Terminal Services' é mais exigente que trabalhar as mesmas horas numa aplicação Web.

### **2.5.4 - Independência**

Nenhuma das soluções contempla este requisito a nível do servidor. Ao nível do cliente apenas o 'Corecon' não exige plataforma Windows, uma vez que também é uma aplicação Web, favorecendo a utilização de dispositivos móveis de última geração.

### **2.5.5 - Utilização em Portugal**

Existem ainda muitas empresas que fazem este tipo de tarefas usando aplicações de folha de cálculo, sobretudo pequenas empresas que não têm estrutura para suportar a curva de aprendizagem necessária para aprender a dominar um software de orçamentação, a par de que, os utilizadores também demonstram alguma resistência à mudança originando alguma falta de motivação.

O Candy / CCS é utilizado na maioria das médias / grandes empresas que utilizam software de orçamentação (“CCS – Clientes principais”, 2011) e quer em conversas com utilizadores, quer lendo os anúncios para orçamentistas, é o software que mais é requisitado, a par do Primavera. Nos software integrados, apesar das promessas, é comum terem problemas de integração, já que essas empresas são na realidade empresas de software de contabilidade e gestão e os módulos de orçamentação são acessórios e nunca funcionam convenientemente porque é necessário criar uma codificação geral para todos os serviços (contabilidade, orçamentação, armazém, compras, etc) e isso acaba por ser impossível, já que as necessidades de cada departamento são diferentes.

## **2.6 - Importância das APIs**

Uma API, ou 'Application Programming Interface', é um conjunto de rotinas e/ou protocolos que fornecem aos programadores funcionalidades pré-definidas. Estas permitem que os programadores possam desenvolver aplicações que utilizam estas funcionalidades, sem terem que as programar de raiz, de modo a que as suas aplicações sejam desenvolvidas com maior rapidez ou que utilizem dados fornecidos por outras aplicações. No passado, as 'API' eram largamente associadas a programação de sistemas operativos ou aplicações tradicionais mas, nos últimos anos, temos vindo a verificar um aumento exponencial da sua utilização em aplicações 'Web'. Deste modo, quando usadas no desenvolvimento 'Web', uma 'API' é tipicamente definida como um conjunto de mensagens HTTP, cujas respostas podem ser, entre outros formatos, XML ou JSON.

### **2.6.1 - Web API (Web Service)**

As WebAPI permitem aos programadores construir páginas e aplicações baseadas na Internet, usando dados de várias fontes 'online'. Apesar de existirem WebAPI para muitos tipos de indústria, algumas das mais populares são largamente utilizadas nas categorias de redes sociais, e-commerce, música e sistemas de autenticação.

### **2.6.2 - O aumento de popularidade das WebAPIs**

No verão de 2005, a ProgrammableWeb, um dos maiores diretórios online de API e recursos, tinha listadas apenas 32 APIs. Em Março de 2011, a mesma fonte chegava já às 3000 APIs (“API Directory – ProgrammableWeb”, 2011).

Em Outubro de 2011 chegava às 4000, isto é, apenas em seis meses aumentava em 30% o total do seu diretório de WebAPIs.

### **2.6.3 - WebAPIs mais populares**

O maior aumento de utilização deve-se, sem sobra de dúvidas, às redes sociais pelo que, as mais populares, são as referentes à área de 'Social Media', entre estas estão os gigantes @Facebook, @Twitter e recentemente o @Google+ cuja API apenas permite leitura mas poderá, e deverá, mudar no futuro.

### **2.6.4 - Protocolos Web API**

Não obstante o protocolo SOAP ou 'Simple Object Access Protocol' ser a norma, o REST ou 'Representational State Transfer' tem sido o protocolo mais utilizado devida à sua facilidade de utilização, sendo, assim, o protocolo 'de facto'.

### **2.6.5 - Formatos de dados Web API**

O formato mais utilizado é, de longe, o XML, no entanto o JSON tem vindo a aumentar consistentemente a sua popularidade e utilização. Muitos programadores preferem o formato JSON devido às seguintes razões:

- Os dados podem ser lidos sem necessidade de interpretadores complexos.
- Os interpretadores são rápidos, pequenos e simples.
- O peso dos dados em JSON é muito inferior, uma vez que o JSON não inclui 'namespaces', 'atributos' e complexa estrutura como o XML.
- Muitos provedores de APIs fornecem ambos os formatos, XML e JSON, mas a popularidade do JSON chegou ao ponto de certos provedores passarem a fornecer exclusivamente em formato JSON.

### **2.6.6 - Benefícios das WebAPIs**

Apesar do público alvo ser, por norma, a comunidade de programadores, as WebAPI podem ser também benéficas para utilizadores finais assim como para os próprios provedores.

- Programadores – Facilitam o uso de funcionalidades avançadas para serem integradas em páginas ou aplicações que criam.
- Utilizadores finais – Uma boa API aumenta a utilidade de páginas e aplicações tornando-as mais interativas e agradáveis.
- Provedores – Uma API bem construída é adotada pela comunidade de programadores e pode ser uma boa ferramenta de marketing uma vez que:
  - Aumenta a lealdade à Marca
  - Aumenta o interesse nos produtos e serviços da Empresa
  - Aumenta os seus visitantes



## 2.7 - Importância das Bibliotecas

Uma Biblioteca é um conjunto de ferramentas ao serviço do programador, que disponibilizam controlos e comportamentos cujo objetivo é melhorar a experiência do utilizador.

Estas podem ser uma coleção de recursos usados para desenvolver aplicações e incluem código e sub-rotinas, classes, valores e especificações de tipos.

As bibliotecas contêm código e/ou dados que fornecem serviços a programas independentes, encorajando a partilha e alteração do código de modo modular.

Para melhor dar a noção da importância das bibliotecas no desenvolvimento de aplicações para a Internet poderemos analisar o seguinte quadro:

#	Biblioteca	Sítios	Utilizadores	Visualizações
1	jQuery	1,142,493	449,878	36,924,114
2	jQuery UI	170,235	187,465	3,945,617
3	Prototype	165,216	174,876	4,085,275
4	MooTools	153,872	142,868	3,442,928
5	YUI	104,476	147,523	4,829,596
6	script.aculo.us	71,454	90,512	1,497,539
7	Modernizr	25,913	41,376	678,155
8	dojo	17,515	18,945	426,286
9	ExtJS	11,213	13,298	291,676
10	xajax	3,607	6,284	107,4
11	MochiKit	1,141	3,264	58,55
12	DHTMLX	696	789	12,343

Fonte: <http://wappalyzer.com/stats/cat/JavaScript%20frameworks>

*Tabela 2: Importância das bibliotecas*

É visível a importância das bibliotecas no desenvolvimento de aplicações, nomeadamente a 'Framework' JQuery que é largamente a mais utilizada.

## 2.8 - Tecnologia Adotada

A tecnologia adotada para desenvolver o Ez-Bud baseou-se unicamente em soluções de código aberto que proporcionam uma relação qualidade-preço imbatível e, por isso mesmo, escalabilidade apenas limitada pela imaginação do programador ou gestor de

sistemas responsável pela manutenção da aplicação.

### 2.8.1 - W3C

O W3C, ou 'World Wide Web Consortium' é uma comunidade internacional onde as organizações que dela fazem parte, uma equipa dedicada e o público trabalham em conjunto para desenvolver normas para a Internet. Liderada pelo inventor da 'Web' Tim Berners-Lee e Jeffrey Jaffe como CEO<sup>2</sup> a missão desta organização é aumentar o potencial da Internet através da especificação de normas que facilita a experiência do programador e dos utilizadores em geral.

### 2.8.2 - HTML

Em 1980, Tim Berners-Lee então a trabalhar no CERN<sup>3</sup>, propôs um sistema para os investigadores do CERN partilhar documentos ao qual chamou 'Enquire'. Em 1989, Tim Berners-Lee escreveu um memorando propondo um sistema de hipertexto baseado na Internet, especificando o HTML ou 'HyperText Markup Language' fazendo também o código de servidor e navegador necessários para todo o sistema funcionar. O HTML é a linguagem predominante para páginas eletrónicas e é escrito com base em elementos, ou 'tags', cuja sintaxe é a seguinte:

- `<h1> cabeçalho </h1>`

No exemplo, vemos o início do elemento 'h1', respetivo final '/h1', pelo que 'cabeçalho' será o texto que está identificado como h1 ou cabeçalho de nível um. Nem todas as 'tags' necessitam de ser finalizadas, como é o seguinte caso:

- ``

Neste caso, estamos a definir uma imagem para que seja incorporada na página. Pode, ainda haver uma estrutura hierárquica onde elementos englobam outros elementos.

Os navegadores podem ainda definir CSS ou 'Cascading Style Sheets' para definir o aspeto dos elementos, posicionamento, etc, sendo esta a prática aconselhada.

---

2 Chief Executive Officer, em português Diretor Executivo

3 Conseil Européen pour la Recherche Nucléaire, em português Organização Europeia para a Pesquisa Nuclear

### 2.8.3 - CSS

CSS é uma linguagem para descrever o modo como o HTML ou XML serão apresentados no ecrã, papel ou qualquer outro media que esteja devidamente especificado. Utiliza propriedades para definir as cores, texto, fundos ou bordas dos elementos presentes numa determinada página.

Quando a primeira especificação foi publicada pelo W3C, todas as especificações estavam num único documento que definiu o CSS de nível 1. As CSS não têm versões mas sim níveis (Cascading Style Sheets (CSS) Snapshot 2010, 2011), sendo que cada nível de CSS complementa o anterior refinando as definições e adicionando funcionalidades, assim, um conjunto de funcionalidades ou comportamentos de cada nível é um supraconjunto do nível subsequente.

As CSS de nível 2 também foram definidas num único documento, composto por vários capítulos ao passo que o nível 3 adotou uma abordagem mais modular onde cada módulo define uma parte da CSS deixando assim de ter uma lógica monolítica. Esta abordagem é mais versátil e manejável e permite mais rapidamente o incremento de melhorias.

Para mais detalhes está presente a especificação em <http://www.w3.org/TR/CSS/> para consulta pública.

Relativamente à compatibilidade entre navegadores apresenta-se o seguinte quadro:

<b>Propriedades</b>	<b>Nível CSS</b>	<b>Internet Explorer</b>	<b>Firefox</b>	<b>Safari</b>	<b>Opera</b>	<b>Google Chrome</b>
Pseudo-elements						
::after	2.1	8	1	1	6	0.2
::before	2.1	8	1	1	6	0.2
::first-letter	1	5.5	1	1	6	0.2
::first-line	1	5.5	1	1	6	0.2
Pseudo-classes						
:active	1	8	1	1	6	0.2
:checked	3	9	1	3.1	9	0.4
:disabled	3	9	1	3	9	0.2
:empty	3	9	1	3	9	0.2
:enabled	3	9	1	3	9	0.2
:first	2	8	-	-	-	-
:first-child	2.1	7	1	3.1	9.5	0.4
:first-of-type	3	9	1	3.1	9.5	0.4
:focus	2.1	8	1	1	7	0.2
:hover	2.1	7	1	1	6	0.2
:lang	2.1	8	1	3	8	0.2
:last-child	3	9	1	3.1	9.5	0.4
:last-of-type	3	9	1	3.1	9.5	0.4
:left	2.1	8	-	-	-	-
:link	1	5	1	1	6	0.2
:not	3	9	1	1	9	0.2
:nth-child	3	9	1	3.1	9.5	0.4
:nth-last-child	3	9	1	3.1	9.5	0.4
:nth-last-of-type	3	9	1	3.1	9.5	0.4
:nth-of-type	3	9	1	3.1	9.5	0.4
:only-child	3	9	1	3.1	9.5	0.4
:right	2.1	8	-	-	-	-
:root	1	9	1	1	9.5	0.2
:visited	1	5	1	1	6	0.2
only-of-type	3	9	1	3.1	9.5	0.4
Posicionamento						

Propriedades	Nível CSS	Internet Explorer	Firefox	Safari	Opera	Google Chrome
bottom	2.1	8	1	1	6	0.2
clip	2.1	5.5	1	1	6	0.2
display	1	8	1	1	7	0.2
elevation	2.1	-	-	-	-	-
height	1	5	1	1	6	0.2
left	2.1	8	1	1	6	0.2
max-height	2.1	7	1	2	6	0.2
max-width	2.1	7	1	2	6	0.2
overflow	2.1	7	1	1	7	0.2
position	2.1	7	1	1	6	0.2
right	2.1	8	1	1	6	1
top	2.1	8	1	1	6	0.2
visibility	2.1	5	1	1	6	0.2
width	1	5	1	1	6	0.2
word-spacing	1	8	1	1	6	0.2
z-index	2.1	8	1	1	6	0.2

*Tabela 3: Compatibilidade de selectors CSS e principais navegadores*

Nota: Como versão considera-se em conta o cumprimento total da norma

Como podemos verificar por este subconjunto de 'Selectors' a compatibilidade entre navegadores é, nalguns casos, muito díspar (Browser Support of CSS Properties & Selectors, 2011) o que faz com que o programador tenha que ter o cuidado de verificar estas diferenças e, muitas vezes, organizar os seus estilos de forma a ser compatível com qualquer navegador.

## 2.8.4 - HTML5

Nos últimos anos, conceitos como a Web 2.0, 'RIAs' ou 'Rich Internet Applications' e a Web semântica pressionaram o HTML, CSS e Javascript para além dos seus limites, muitas vezes relegando algumas funcionalidades para 'plugins' como o ®Adobe Flash para a obtenção de um interface mais rico e interativo para o utilizador final. O ®Adobe FLEX, ®Microsoft Silverlight e ®JavaFX têm provido suporte onde o HTML se revela escasso.

Como objetivos principais do HTML5, em desenvolvimento pelo W3C (HTML5, 2011), podemos enumerar os seguintes:

- Modernizar o HTML
- Melhorar o acesso ao DOM ou 'Document Object Model'
- Retrocompatibilidade
- Fazer com que as implementações cumpram as especificações
- Especificações claras e inequívocas

Assim o surgimento do HTML5, com suporte multimédia, armazenamento de dados local, suporte para aplicações 'Offline' e suporte para desenho 2D nativo vem provar que com HTML, CSS e Javascript se conseguem desenvolver aplicações ricas. As últimas especificações do HTML e CSS apontam para a prescindível utilização de 'plugins' para a construção destas aplicações ricas assim como a redução do Javascript necessário. O HTML5 ainda está em fase de discussão se bem que existe já algum suporte nos navegadores mais recentes. Estes podemos enumerar, mas não está limitado aos seguintes:

- Mozilla Firefox - versão 3.5 ou superior
- Apple Safari - versão 4.0 ou superior
- Opera - versão 10.0 ou superior
- Google Chrome - versão 3.0 ou superior
- Internet Explorer - versão 9.0 ou superior

### 2.8.4.1 - Novas funcionalidades do HTML5

As novas funcionalidades que o HTML5 fornece não são definitivas, pois como anteriormente referido, é um trabalho em curso. Resumindo as novas funcionalidades enumeram-se as principais áreas de estudo que já têm algum suporte pelos principais navegadores.

- Elementos semânticos
- Canvas
- Video
- Audio
- Formulários
- Armazenamento local

## Elementos semânticos

A especificação inclui uma série de novos elementos semânticos que são usados para dar algum significado a algumas áreas de uma página, tais como o cabeçalho, rodapé, navegação, etc. Em versões prévias de HTML são, por norma 'de facto', usados elementos '<div>' para criar estas áreas da página, usando o atributo 'id' e 'class' para os diferenciar. O problema desta abordagem é que não têm um significado semântico fazendo com que seja muito difícil para o código, principalmente o próprio navegador, saber o que cada área em particular está a fazer ou o seu significado. Obviamente poderá continuar a usar-se esta abordagem mas não é recomendável para tirar todo o partido da futura Internet.

Os principais elementos semânticos no HTML5 são os seguintes:

- <header> - Este elemento é usado para definir o cabeçalho de um determinado elemento, seja ele a própria página, um artigo ou uma secção.
- <footer> - De igual forma ao anterior, define o rodapé dum elemento
- <nav> - Este contentor é para a área principal de navegação da página.
- <article> - Define uma área independente na página que pode ser distribuída por si só, tais como notícias ou comentários, que são tipicamente distribuídas por 'RSS Feeds'
- <section> - Este elemento representa uma secção de uma página, tal como um capítulo ou secção de um artigo.
- <aside> - Este novo elemento pode ser usado para definir uma barra lateral, ou outro conteúdo que pode ser considerado à parte do conteúdo principal da página, como por exemplo publicidade ou menus auxiliares.
- <hgroup> - Em alguns casos, uma página, artigo ou secção pode ter necessidade de ter mais que um título ou uma hierarquia como título e subtítulo.

## O elemento <CANVAS>

Este elemento foi inicialmente desenvolvido pela ®Apple para os elementos de ambiente de trabalho ou 'Dashboard widgets' do ®Mac OS X e no navegador 'Safari'. Posteriormente foi adotado também pela ®Mozilla, ®Opera e ®Google Chrome nos seus navegadores, assim foi incluído no HTML5 juntamente com uma serie de APIs de desenho 2D que pode ser usada para criar formas, texto, transições e animações dentro do elemento. Muitos programadores acham que este elemento é uma das principais novidades e que facilita a produção de gráficos, jogos e aplicações gráficas sem necessidade de 'plugins' externos.

A sua utilização é básica sendo apenas definida a sua altura, largura e identificador para o objeto, onde o programador poderá manusear elementos criados por código em Javascript, tipicamente quando a página termina o seu carregamento. O programador poderá então criar formas, aplicar cores, opacidades, gradientes, criar texto e transformar o próprio objeto para desempenhar animações. As APIs também possibilitam que possa haver interatividade e responder a comandos do utilizador tais como eventos, rato e teclado, facilitando a produção de jogos e aplicações.

## Os elementos <audio> e <video>

A popularidade dos meios de áudio e vídeo tem, inequivocamente, vindo a crescer de forma exponencial. Infelizmente, a norma atual, não foi construída para a difusão deste tipo de média e, em resultado disso mesmo, estes conteúdos têm vindo a ser distribuídos utilizando essencialmente o formato ®Flash Video (.flv) e consequentemente a utilização do 'plugin' ®Adobe Flash Player. No entanto, o HTML5 inclui suporte para dois novos elementos, <audio> e <video>, que permitem ao programador incluir conteúdo multimédia sem a necessidade de ter que obrigar o utilizador a ter que instalar 'plugins'. Teoricamente tem uma vantagem chave ao consumir menos recursos de CPU e, logo, consumir menos energia.

Por outro lado, um assunto ainda em debate prende-se com os formatos suportados por cada navegador e respetivo licenciamento de patentes que cada CODEC<sup>4</sup> pode obrigar. Assim, de momento, Mozilla e Opera estão a suportar o Codec Theora por ser de código aberto e, por isso, sem necessidade de licenciamento de patentes, ao passo que a Apple e Google preferem o Codec H.264, tipicamente usados em ficheiros do tipo MP4,

---

4 Codificador/Descodificador



MOV ou MKV, por razões de desempenho nomeadamente no que diz respeito a difusão de conteúdos em alta definição. Este problema não se resume ao vídeo, pois relativamente ao áudio existe um problema semelhante uma vez que os formatos MP3 e AAC têm patentes e o formato Vorbis não. Neste campo, o formato Vorbis tem um grande contra pois toda a indústria, praticamente, se baseia no formato MP3 ou não possui suporte para Vorbis.

Há ainda muitas decisões a tomar quanto a estes dois elementos num futuro próximo, entretanto, o programador, pode começar por dar suporte a todos estes formatos.

Até lá esperemos que a decisão final não seja serem os navegadores a definirem qual o formato que vão suportar pois retiraria muita versatilidade a estes novos elementos.

### **Melhorias nos formulários**

Os controlos de entrada, ou `<input>`, usados no HTML são os seguintes:

- `button` – botões para interação por parte do utilizador
- `submit` – Define um botão de submissão para enviar dados para o servidor
- `image` – Define uma imagem como botão de submissão
- `file` – Para selecionar ficheiros
- `reset` – Reinicia todos os valores do formulário para o seu estado inicial
- `checkbox` – Define uma caixa de seleção múltipla
- `radio` – Define uma caixa de seleção única
- `text` – Define uma caixa de texto
- `password` – Define uma caixa de texto mas com os caracteres de entrada não legíveis.
- `hidden` – Define um campo escondido auxiliar

Adicionalmente, há outros controlos como o `<select>` ou `<textarea>` para caixas de seleção de valores pré-definidos ou entrada de texto multi-linha respetivamente.

Estes controlos disponibilizam quase todas as formas de entrada conhecidas, mas, à medida que a Internet foi evoluindo foram surgindo outras necessidades, como escolha de datas, caixas de seleção de cor, deslizadores ou 'sliders', etc. Para colmatar estas necessidades, os programadores foram construindo, através de código de Javascript ou, mais uma vez, recorrendo aos 'plugins' ®Adobe Flex, ®Microsoft Silverlight, ou ®JavaFX.

O HTML5 tenta assim colmatar estas necessidades fornecendo um novo conjunto de tipos de controlo:

1. `Color` – Define um campo validando que deve ser uma cor

2. Date – Define um campo cuja entrada deve ser uma data
3. Datetime – Define um campo cuja entrada deve ser uma data e hora 'UTC'
4. Datetime-local – Define um campo cuja entrada deve ser uma data e hora local
5. Month – Define um campo cuja entrada deve ser mês
6. Time – Define um campo cuja entrada deve ser uma hora
7. Week – Define um campo cuja entrada deve ser uma semana
8. Number – Define um campo cuja entrada deve ser um número
9. Range – Define um campo cuja entrada deve ser uma hora
10. Search – Define um campo usado como campo de pesquisa
11. Email – Define um campo cuja entrada deve ser um e-mail
12. Tel – Define um campo cuja entrada deve ser um número de telefone
13. Url – Define um campo cuja entrada deve ser um URL<sup>5</sup>

Relembro que a especificação está a ser definida e que, por enquanto, há diferentes implementações destes controlos, por exemplo, o campo tipo 'color' no Opera é um campo em que surge uma janela auxiliar que permite a seleção duma cor, ao passo que no Google Chrome apenas valida a entrada de um valor hexadecimal. É, pois, uma área que ainda está a ser definida e ainda muito limitada. Adicionalmente, nestes novos controlos, o HTML5 suporta duas novas funcionalidades:

- 'autofocus' – Quando a página é construída, o campo que estiver definido com este atributo terá o foco automático, não sendo necessário fazê-lo com Javascript.
- 'placeholder' – Tem como objetivo mostrar texto por omissão quando o campo estiver vazio.

#### **2.8.4.2 - Interfaces de programação**

##### **Armazenamento**

Os programadores, tradicionalmente, têm usado 'cookies' para armazenar informação do utilizador, numa determinada máquina, permitindo que a aplicação a possa recolher, mais tarde, quando necessário. É uma funcionalidade útil mas muito limitada. Por norma, os navegadores não suportam mais que 20 'cookies', por servidor ou 4Kb de dados por 'cookie' e aqui já se inclui o próprio nome além do valor. Adicionalmente, são enviados a cada pedido HTTP o que ocupa imensos recursos. O HTML5 fornece uma solução através da API 'Local Storage' ou armazenamento local, que está coberta por uma especificação à

---

5 Uniform Resource Locator ou Localizador-Padrão de Recursos.

parte da do HTML5. Assim passa a ser possível guardar informação de um utilizador sem utilização de ‘cookies’ e que está sempre acessível, sem necessidade do vai-vem de ‘cookies’ entre o servidor e o cliente nem, tampouco, estar a carregá-la constantemente. Tem, obviamente, restrições no que diz respeito ao domínio para não haver fugas de informação entre domínios.

O armazenamento tem duas formas, ‘sessionStorage’ e ‘localStorage’ cada um fornecendo métodos similares para gerir informação:

- `setItem()`
- `removeItem()`
- `getItem()`

O armazenamento de sessão ou ‘sessionStorage’ tem como objetivo guardar a informação da sessão corrente, isto é, enquanto o navegador não é reiniciado, ao passo que o armazenamento local, ou ‘localStorage’, será para armazenamento persistente. Há ainda eventos que monitorizam e, eventualmente, reagem a atividades de armazenamento.

Existem ainda duas especificações que estão muito relacionadas com o armazenamento:

- ‘indexedDB’ – Ainda num estágio muito inicial tem como intenção disponibilizar ainda mais funcionalidades para armazenamento persistente, disponibilizando métodos para questionar a base de dados de forma mais ágil ou guardar objetos complexos não apenas cadeias de caracteres ou ‘Strings’.
- ‘Web Workers’ – Permitirá aos programadores desenhar blocos de programação que serão executados de forma assíncrona, permitindo assim executar operações complexas sem bloquear o processamento do navegador.

### **Aplicações em modo desligado ou ‘offline’**

A maioria dos navegadores armazenam as páginas em memória física local ou ‘cache’ permitindo que seja possível visualizá-las mesmo sem ligação ativa, isto é em modo ‘offline’. Isto funciona muito bem para páginas estáticas, mas não quando são fruto de código de servidor ou dinâmicas que acedem a bases de dados. O HTML5 fornece suporte para aplicações em modo desligado, onde o navegador descarrega todas as páginas necessárias, definidas pelo programador, ao modo de funcionamento desligado e assim, quando o utilizador estiver sem conectividade a aplicação funcionará, teoricamente, de igual modo. Assim que regressar a conectividade o navegador sincronizará com o servidor.

## **Edição em linha**

O HTML5 definiu uma nova API para facilitar a edição de conteúdos. Embora existam inúmeras soluções em Javascript que o facilitam, a disponibilização pelo HTML5 ajudará o utilizador a ter uma experiência mais consistente.

## **Ficheiros**

A API 'file' estabelece a especificação para trabalhar com ficheiros. Possibilita ter acesso ao próprio conteúdo do ficheiro no navegador.

## **Arrastar e largar**

A nova API para arrastar e largar, ou 'drag-and-drop', diminui, uma vez mais, a quantidade de código Javascript necessário a funcionalidades básicas. Juntamente com a API para ficheiros é possível, pelo utilizador, arrastar ficheiros do computador diretamente para o navegador, como sucede no @Gmail tanto no @Mozilla Firefox como no @Google Chrome.

## **Histórico**

Numa era em que as aplicações tiram o máximo proveito da tecnologia AJAX, o histórico de visitas deixa de fazer sentido uma vez que, como as páginas são carregadas parcialmente, o histórico tradicional não toma este novo paradigma em consideração, pelo que em algumas aplicações, o histórico é inexistente ou inconsistente. Mais uma vez, os programadores têm ultrapassado esta questão com auxílio a código Javascript, mas se for um comportamento nativo do navegador passará a ser, sem dúvida, uma enorme mais valia.

## **Microdata**

Por vezes há a necessidade de anotar conteúdo específico aos elementos permitindo uma análise semântica do próprio elemento. Assim os programadores poderão usar 'microdata' para agrupar pares de chave-valor nas suas páginas.

## **SVG**

SVG significa 'Scalable Vector Graphics' e é uma linguagem para descrever imagens 2D e aplicações gráficas em XML que por sua vez é interpretado por um visualizador SVG. É muito útil para diagramas gráficos.

## **Localização Geográfica**

Esta API fornece uma interface para detetar a localização geográfica de um dispositivo. A informação sobre a localização é detetada através de uma série de fontes, incluindo endereço IP, dispositivo GPS, rede sem fios, entre outras.

## **'WebSocket'**

Permite ter fluxos bidirecionais entre o navegador e o servidor através de uma ligação TCP. Ideal para o desenvolvimento de jogos tendo, já, suporte nos navegadores mais recentes ®Firefox, ®Opera, ®Chrome, e ®Safari.

## **Eventos de servidor**

Esta especificação permitirá ter uma ligação HTTP para receber notificações enviadas por servidores, na forma de eventos DOM. Será, por assim dizer, o inverso do paradigma atual onde os navegadores é que consultam o servidor para atualizações, reduzindo o dispêndio de recursos de largura de banda e processamento em desnecessários pedidos ao servidor, tendo apenas que receber o que o servidor enviar.

## **WebGL**

A API WebGL ou 'Web-based Graphics Library', possibilitará a criação de gráficos em três dimensões e interativos no navegador. Atuará no contexto do objeto 'canvas'.

### **2.8.5 - JAVASCRIPT**

O Javascript foi inicialmente desenvolvido por Brendan Eich e implementado em 1995 pela Netscape Communications Corporation no navegador Netscape Navigator 2 beta, sob o nome 'Mocha', mais tarde LiveScript e finalmente Javascript. A Microsoft fez também uma linguagem a que chamou JScript sob os mesmos fundamentos mas evitando assim problemas de licenciamento. Em 1996, a Netscape submeteu o Javascript à ECMA International para que fosse considerado norma e ao resultado dessa análise e trabalho subsequente foi chamado ECMAScript, sendo neste momento a norma (ECMA International, 2011).

O Javascript é uma linguagem orientada a objetos com herança por 'prototype', isto é, não existem classes mas a herança é transmitida por clonagem. Permite, além dos objetos pré-definidos, que o programador possa criar ou remover os seus.

O Javascript é uma linguagem interpretada, sendo que atualmente os navegadores mais recentes suportam compilação JIT, assim o navegador cliente interpreta o código, analisa e executa-o imediatamente, ou pré-compila-o para otimizar o desempenho (Goodman, 2001).

A compilação JIT melhora significativamente o desempenho e, obviamente, a experiência do utilizador em aplicações mais complexas. O Javascript é, sem dúvida, uma das ferramentas mais populares dos programadores que desenvolvem aplicações para a Internet.

### 2.8.6 - PHP

O desenvolvimento do PHP começou em 1994 por Rasmus Lerdorf (PHP Documentation Group, 2011), que criou um conjunto de 'scripts' escritos em PERL<sup>6</sup>, a que chamou 'Personal Home Page Tools', para fazer a manutenção à sua página pessoal que apenas mostrava o seu Curriculum Vitae e registava o tráfico da página. Já em 1997 Zeev Suraski e Andi Gutmans rescreveram o interpretador que serviu de base para o PHP3, sendo que neste caso PHP já significava Hypertext Preprocessor sendo um acrónimo recursivo. Neste momento o PHP5 é a versão oficial.

As inúmeras possibilidades do PHP, ser uma tecnologia de código aberto e uma enorme comunidade de utilizadores fez do PHP uma das mais populares linguagens de programação para aplicações para a Internet como podemos ver na Figura 10:

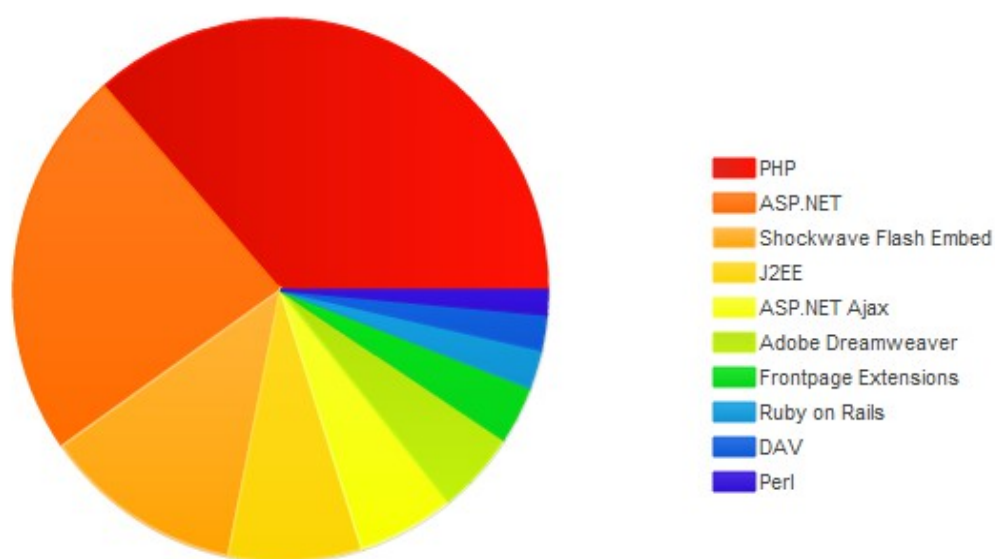


Figura 2: Linguagens de programação utilizadas nos 10000 sites mais visitados, por Builtwith

O PHP é uma tecnologia versátil, funcionando na esmagadora maioria de sistemas operativos, servidores de páginas e disponibiliza ligações para, eventualmente, todas as bases de dados. Todos estes fatores, assim como o ótimo desempenho, são muito importantes para a escolha do PHP para qualquer tipo de projeto. Os mais conhecidos

<sup>6</sup> Perl foi desenvolvido por Larry Wall em 1987 para sistemas Unix para utilização genérica

projetos de código aberto tem base em PHP desde o ®Wordpress, ®Joomla e até mesmo o ®Facebook. Orientada a objetos, ou não, é sem dúvida a linguagem de programação mais versátil para o desenvolvimento de aplicações para a Internet, sendo mesmo possível usá-la em linha de comando e até em ambientes gráficos através de GTK (PHP-GTK: Documentation, 2011).

### Como funciona

Desde que num servidor capaz de interpretar páginas PHP o programador poderá intercalar código HTML com código PHP da seguinte forma:

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php
      echo "Olá Mundo!!!";
    ?>
  </body>
</html>
```



### 2.8.7 - MYSQL

O MySQL é o mais popular Sistema de Gestão de Base de Dados Relacional, ou SGBDR, em inglês RDBMS, e é desenvolvido, distribuído e suportado pela ®Oracle Corporation. O modelo relacional é um modelo de dados, que se baseia no princípio em que todos os dados estão guardados em tabelas ou, matematicamente falando, relações. Toda sua definição é teórica e baseada na lógica de predicados e na teoria dos conjuntos.

O conceito foi criado por Edgar Frank Codd em 1970, sendo descrito no artigo ‘Relational Model of Data for Large Shared Data Banks’. Na verdade, o modelo relacional foi o primeiro modelo de dados descrito teoricamente, os restantes modelos são conhecidos como modelo hierárquico, modelo em rede ou ‘Codasyl’ e modelo de listas invertidas.

Em 1985, Edgar Frank Codd, publicou um artigo onde definia 13 regras para que um Sistema de Gestão de Base de Dados, ou SGBD seja considerado relacional (Codd, 1985):

1. Regra Fundamental - Um SGBDR deve gerir os seus dados usando apenas suas capacidades relacionais.
2. Regra da informação - Toda informação deve ser representada de uma única forma, como dados em colunas e linhas numa tabela.
3. Regra da garantia de acesso – Qualquer valor pode ser acessado logicamente e unicamente, usando o nome da tabela, o valor da chave primária da linha e o nome da coluna.
4. Tratamento sistemático de valores nulos - Os valores nulos, diferente do zero, da cadeia de caracteres vazia, da cadeia de caracteres em branco e outros valores não nulos, existem para representar dados não existentes de forma sistemática e independente do tipo de dado.
5. Catálogo dinâmico on-line baseado no modelo relacional - A descrição da base de dados é representada no nível lógico como dados ordinários, isto é, em tabelas, permitindo que utilizadores autorizados as manipulem como se de dados se tratasse.
6. Regra da sub-linguagem compreensiva - Um sistema relacional pode suportar várias linguagens e formas de uso, porém deve possuir pelo menos uma linguagem com sintaxe bem definida e expressa por cadeia de caracteres e com habilidade de apoiar a definição de dados, a definição de vistas, a manipulação de dados, as restrições de integridade, a autorização e a fronteira de transações.
7. Regra da atualização de vistas - Toda a vista, ou ‘view’ que for teoricamente atualizável será também atualizável pelo sistema.

8. Inserção, atualização e eliminação de alto nível - A capacidade de manipular a tabela base ou tabelas derivadas como um operador único não se aplica apenas à recuperação de dados, mas também a inserção, alteração e eliminação de dados.
9. Independência dos dados físicos - Programas de aplicação ou atividades de terminal permanecem logicamente inalteradas quaisquer que sejam as modificações na representação do armazenamento ou métodos de acesso internos.
10. Independência lógica de dados - Programas de aplicação ou atividades de terminal permanecem logicamente inalteradas quaisquer que sejam as mudanças de informação.
11. Independência de integridade - As relações de integridade específicas de uma base de dados relacional devem ser definidas e armazenadas no catálogo. Deve ser possível alterar restrições sem afetar Programas de aplicação ou atividades de terminal.
12. Independência de distribuição - A linguagem de manipulação de dados deve possibilitar que as aplicações permaneçam inalteradas estejam os dados centralizados ou distribuídos fisicamente.
13. Regra da Não-subversão - Se o sistema relacional possui uma linguagem de baixo nível, não deve ser possível subverter ou ignorar as regras de integridade e restrições definidas no alto nível.

Uma base de dados relacional armazena dados em tabelas separadas e não todos os dados no mesmo local. Isto aumenta exponencialmente o desempenho e flexibilidade.

A linguagem mais usada para aceder a dados é o SQL, ou 'Structured Query Language', estando definida como ANSI/ISO SQL.

Ainda que mantido pela ®Oracle<sup>7</sup>, o MySQL continua a ter uma versão de código aberto e a versão de comunidade pode ser utilizada de forma gratuita, não tendo, obviamente direito a qualquer tipo de suporte pela ®Oracle mas apenas da comunidade que é das maiores comunidades mundiais. O licenciamento baseia-se na licença GPL. Para os projetos que não cumpram os requisitos da licença GPL podem optar pelo licenciamento comercial.

---

7 ®Oracle é uma marca registada.

### 2.8.8 - Formato MPX

O formato de ficheiro MPX 4.0 é um formato de ficheiro ASCII que pode utilizar para transferir informações do projeto entre diferentes versões do @Microsoft Project, bem como programas de outros fabricantes que suporte o formato de ficheiro MPX.

O ficheiro de formato de ficheiro MPX descreve o formato de ficheiros MPX 4.0 criados no @Microsoft Project, versões 4.0, 4.1 e 98. Pode-se encontrar estes ficheiros na pasta onde se instalou o @Microsoft Project versão 4.0 ou 4.1.

Exemplo de ficheiro com o formato MPX:

```
MPX,Microsoft Project for Windows,4.0,ANSI
10,$,1,2,"",,.
11,2,0,2,8.00,40.00,$0.00/h,$0.00/h,1,1
12,0,0,480,/,:AM,PM,9
20,Standard,0,1,1,1,1,1,0
25,1
25,2,8:00 AM,12:00 PM,1:00 PM,5:00 PM
25,3,8:00 AM,12:00 PM,1:00 PM,5:00 PM
25,4,8:00 AM,12:00 PM,1:00 PM,5:00 PM
25,5,8:00 AM,12:00 PM,1:00 PM,5:00 PM
25,6,8:00 AM,12:00 PM,1:00 PM,5:00 PM
25,7
30,"Orçamento 15..","David Pereira","David Pereira",Standard,"Thu 10/06/2011"
60,"ID","Unique ID","Name","Duration","Fixed","Outline Level","Start","Finish"
61,90,98,1,40,80,3,50,51
70,1,376,"1",23d,No,0,Tue 10/04/2011,Wed 10/26/2011
70,2,379,"1.1",23d,No,1,Tue 10/04/2011,Wed 10/26/2011
70,3,378,"1.2",18d,No,1,Wed 10/05/2011,Sat 10/22/2011
70,4,377,"1.2.1 - fornecimento de coisas",14d,No,2,Sun 10/09/2011,Sat 10/22/2011
70,5,384,"1.2.2",14d,No,2,Wed 10/05/2011,Tue 10/18/2011
70,6,380,"2",0d,No,0,Sun 12/04/2011,
```

## **2.9 - APIs utilizadas**

### **2.9.1 - @Google Apps**

O @Google Apps<sup>8</sup> é um conjunto de ferramentas de produtividade inteiramente online e usado a partir de qualquer navegador ou dispositivo móvel, e que deverá ser parametrizado para um domínio à escolha do futuro utilizador, leia-se administrador, uma vez que só os administradores de domínio poderão configurar um domínio no 'Google Apps'.

Este conjunto de aplicações incluem as ferramentas que servem de suporte a qualquer organização como processamento de texto, folhas de cálculo, apresentações, calendário, comunicações síncronas e assíncronas como o e-mail e mensageiro instantâneo, vulgo 'chat', videoconferência e partilha de ficheiros. De seguida enumero algumas características destes serviços:

#### **2.9.1.1 - Gmail**

- Correio Eletrónico (E-mail) com alojamento de 10Gb (Versão Standard e Educação) ou 25Gb (Versão Empresa)
- E-mail parametrizado para domínio do utilizador
- Proteção contra correio não solicitado (SPAM)
- Anti-vírus
- Pesquisa de E-mail
- Acesso a partir de qualquer computador ou dispositivo móvel
- Sincronização com @Microsoft Outlook
- Suporte via IMAP<sup>9</sup> e POP<sup>10</sup> com encriptação SSL/TSL

#### **2.9.1.2 - Talk**

- Comunicação de mensagens instantâneas de texto, voz e vídeo
- Acesso a partir de qualquer computador ou dispositivo móvel

---

<sup>8</sup> O @Google é uma marca registada.

<sup>9</sup> Internet Message Access Protocol é um protocolo de nível aplicação que permite a um cliente a consulta de e-mail a um servidor de e-mail remoto. Um servidor de IMAP está disponível, por norma, no porto 143.

<sup>10</sup> Post Office Protocol é um protocolo de nível aplicação que permite a um cliente a recolha de e-mail de um servidor de e-mail.

#### **2.9.1.3 - Calendar**

- Calendários privados e partilhados
- Reuniões empresariais, tarefas e coordenação de eventos
- Sincronização com ®Microsoft Outlook, ®Blackberry e outros dispositivos através de aplicações específicas

#### **2.9.1.4 - Docs**

- Processamento de texto, folhas de cálculo, apresentações e repositório de ficheiros com possibilidade de edição por parte de vários utilizadores em simultâneo.
- Acesso 'Offline' aos documentos
- Importação e Exportação de documentos em formato ®Microsoft Office
- Controlo de versões de documentos
- Gráficos

#### **2.9.1.5 - Sites**

- Criação de sítios Internet ou Intranet para partilha de documentos, procedimentos

#### **2.9.1.6 - Video**

- Como um ®Youtube privado é um repositório de vídeo para partilha dos mesmos.
- Alojamento de 3GB para a versão Empresa e de 10GB para a versão Educação

#### **2.9.1.7 - APIs**

- Interfaces de programação para as versões Empresa e Educação

#### **2.9.1.8 - Single Sign On**

- Suporte de autenticação SAML 2.0 para as versões Empresa e Educação. É uma norma baseada em XML para troca de autorizações e autenticações entre domínios, isto é, entre um provedor de identidades e um provedor de serviços. É um produto do comité técnico de serviços de segurança OASIS.

#### **2.9.1.9 - Porquê ®Google Apps?**

A opção por adotar a plataforma ®Google Apps deveu-se ao facto de ser a plataforma em nuvem mais adotada, favorecendo assim a distribuição do Ez-Bud. Neste momento o ®Google Apps alberga mais de dois milhões de organizações, envolvendo cerca

25 milhões de utilizadores que podem descobrir e instalar as aplicações integradas no conceito “Nuvem” que é o próprio @Google Apps. Assim os programadores podem usar o @Google Apps Marketplace para listar e vender os seus serviços e aplicações disponíveis para os utilizadores do @Google Apps. O @Google Apps Marketplace é, assim, uma mostra de aplicações que podem ser usadas por todos os utilizadores do @Google Apps desde que devidamente instaladas pelo administrador do @Google Apps.

### **Tipo de listagem e aplicações**

O Google Marketplace suporta dois tipos de listagens:

- Listagem ‘geral’ – Na listagem os utilizadores são direccionados para o sítio do fornecedor da aplicação para informações e instalação.
- Listagem ‘Instalável’ - Permite ao utilizador instalar a aplicação diretamente do Google Marketplace usando o botão ‘Add it now’ (Adicionar agora) de acordo com a seguinte imagem:

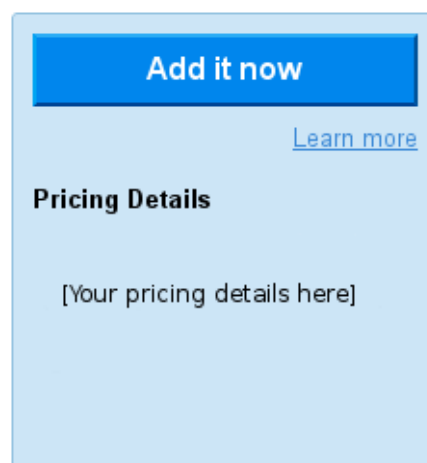


Figura 3: Caixa para adicionar a aplicação a um domínio @Google Apps

### **Requisitos e política de utilização para criação de uma “listagem”**

1. Integração “Single Sign-On”
2. Respeitar a política de aprovação
3. Aceitar a política de pagamento
4. Cumprir os requisitos técnicos.

#### **2.9.2 - OpenID**

OpenID é um sistema de ‘Single Sign-On’ que permite que o utilizador tenha controlo sobre a transação de ‘login’ e respetivos dados online. A simplicidade do sistema permite que um indivíduo tenha apenas um ‘login’ para inúmeros sítios desde que estes permitam ‘login’ seguindo as diretivas OpenID. O OpenID é um sistema descentralizado pelo que garante aos utilizadores não estarem reféns de um qualquer serviço ou ‘Provider’ e é, por natureza, um serviço gratuito.

O que é o OpenID	O que o OpenID não é
É uma prova de identidade, uma forma de provar que nós somos quem dizemos que somos, se houver entrega de credenciais em presença do utilizador.	Não é um sistema com informações fidedignas.
É usado para fazer login sem necessidade de abertura de contas e inserção dos mesmos dados em todo lado	Não é o Big Brother. Não guarda o que se faz onde se faz o login. Isto é controlado individualmente por cada sítio.
É diferente!	Não é complicado! É como andar de bicicleta.
É seguro!	Não é a única solução. Garantir que o fornecedor OpenID é de segurança e houver real prova de Identidade.

### Como Funciona?

De forma genérica, fornecendo o OpenID ao fornecedor do serviço (Relying Party - RP) a que quer aceder, o utilizador é redirecionado para o seu fornecedor OpenID (Provider – IdP) para verificação de credenciais pelo que estas nunca são conhecidas pela RP o que permite, além da simplicidade do sistema, uma maior segurança no que diz respeito ao fornecimento de credenciais online.

Assim, do ponto de vista do utilizador, é um sistema transparente; o utilizador está ligado ao @Google Apps, na sua Suite de colaboração a fazer as suas tarefas diárias e ao aceder à aplicação do Google Marketplace imediatamente é feita a autorização de forma transparente e o utilizador pode começar a trabalhar imediatamente sem ter necessidade de inserção manual de credenciais de qualquer tipo.



- O utilizador invoca a aplicação do 'Google Marketplace' no menu de Navegação Universal
- O utilizador não faz absolutamente nada a não ser esperar que surja a aplicação solicitada.
- O utilizador começa a trabalhar imediatamente na aplicação após solicitá-la.

## Segurança

Existe algum perigo de 'phishing' para os utilizadores menos alerta pois um sítio mal intencionado pode, supostamente, redirecionar o utilizador para outro local 'cópia' do seu 'IdP' sem que este se aperceba e aí fornecer as credenciais o que pode comprometer a segurança destas. Por outro lado, a RP nunca pode confiar cegamente nos dados já que o utilizador pode ter inúmeras identidades e estas não são validadas de forma a garantir a veracidade dos dados. No caso do @Google Apps este problema não se coloca uma vez que a plataforma @Google Apps é um fornecedor fidedigno e a nossa aplicação idem.

## Integração 'Single Sign-On'

A aplicação deve usar sempre Single Sign-On (SSO) e nunca apresentar uma página de login ao utilizador solicitando credenciais. O SSO é uma funcionalidade de segurança obrigatória que deverá estar devidamente implementada na aplicação a listar no @Google Apps Marketplace, sob pena de não ser autorizada a respetiva listagem. Esta obrigação tem como objetivo tanto segurança como conveniência para os utilizadores pois permite que as aplicações façam o utilizador sentir que a aplicação faz parte da Suite @Google Apps levando a uma melhor experiência de utilização. Por outro lado retira ao utilizador o peso da gestão de novas palavras-passe e ao administrador o peso da gestão de utilizadores uma vez que estes já se encontram na Suite @Google Apps. As credenciais dos utilizadores ficam alojados numa localização segura, o provedor "OpenId" o que faz com que o sistema seja mais seguro do que ter várias credenciais de acesso para cada aplicação, ou ainda pior, ter as mesmas credenciais para todas as aplicações ou contas que utiliza.



### **Respeitar a política de aprovação**

A aplicação deve respeitar todos os requisitos para ser aceite no @Google Apps Marketplace. Após criar o perfil de vendedor do Google Marketplace e pagar a taxa aplicável poderá publicar um ou mais anúncios que estarão disponíveis para o público desde que estes adiram à política de aprovação:

- O anúncio e a página do produto devem obedecer às políticas de conteúdo referidos nas diretivas da marca (Google branding guidelines) e ao acordo de distribuição do desenvolvedor (Google Apps Marketplace Developer Distribution Agreement).
- O anúncio deve ser orientado para o negócio ou empresas.
- Se o anúncio publicita:
  - um produto que é uma aplicação web instalável deve suportar OpenId.
  - um produto que seja instalado no cliente ou seja equipamento ou não seja uma aplicação web instalável deve usar uma ou mais APIs Google para integração com o @Google Apps.
  - um serviço, este deve incidir sobre o @Google Apps.

Caso o anúncio respeite os anterior pontos 1,2 e 3 o Google marcará a lista como “Aprovada” sendo o anúncio tornado público, caso contrario será enviada uma explicação por e-mail ao fornecedor.

### **Aceitar a política de pagamento**

Pagamento único, não reembolsável da taxa aplicável (USD \$100) assim como partilha de rendimentos das vendas na ordem dos 20% por cliente nos seguintes casos:

- Rendimentos oriundos de instalações assim que se iniciarem os pagamentos, por exemplo após um período gratuito.
- Se existe uma cobrança periódica, será sempre solicitado uma partilha de rendimentos.
- Rendimentos oriundos de itens cujo preço é por unidade de utilização, por exemplo por Gigabyte, por tempo, etc.
- Rendimentos por Atualizações ou melhorias no produto incluindo:
  - Planos de Atualização como “standard” ou “premium”
  - Utilizadores adicionais
  - Novas funcionalidades

O rendimentos não sujeitos a partilha de rendimentos incluem:

- Rendimentos já existentes.
- Rendimentos oriundos de publicidade.
- Avenças de serviços como instalações, consultoria ou suporte técnico.
- Produtos de comércio eletrónico (bilhetes, reservas, livros, etc).
- Rendimentos de clientes que já não são subscritores do ®Google Apps.
- Taxas

Permanecem gratuitas as APIs gratuitas do Google, gadgets, e o serviço OpenID

### **Cumprir todos os requisitos técnicos**

Os requisitos centram-se essencialmente em garantir uma boa experiência por parte dos utilizadores. Para facilitar o Google fornece a seguinte lista de verificação:

- OpenID e Single Sign-On
  - Suportar OpenID com as contas ®Google Apps
  - Whitelist domínio OpenID. Por norma em aplicações não Google Marketplace as aplicações perguntam por confirmações ao utilizador sempre que a aplicação faz pedidos OpenId ao Google. No entanto uma vez que as aplicações são instaladas por um administrador de domínio este passo deve ser relegado pois assume-se que o administrador do domínio tem autoridade para garantir acesso em nome dos utilizadores, assim ao invés da página de confirmação, a aplicação deve fazer uma verificação do parâmetro openid.realm e compará-la ao declarado no manifesto da aplicação. Sempre que coincidam exatamente o utilizador deve ser autorizado sem confirmação.
  - Permitir tanto pedidos “GET” como “POST” do Google OpenID
  - Suportar que os utilizadores possam aceder diretamente à aplicação sem ter que aceder ao ®Google Apps.
- Acesso a dados
  - Usar o protocolo HTTPS ao aceder a APIs Google Data e OpenID
  - Definir corretamente o âmbito no acesso a dados no manifesto da aplicação.
  - Garantir que a instância de produção está configurada usando a chave correta fornecida.
- Configuração e experiência do utilizador
  - Aplicações que necessitem de configurações adicionais devem retornar o

utilizador para o processo de instalação após as mesmas estarem definidas.

- Não esquecer os atuais clientes tal como os novos. No caso de existirem outras aplicações, a aplicação deve permitir a possibilidade de rapidamente associar os utilizadores e dados com o domínio @Google Apps.

### 2.9.3 - Zend

@Zend é uma 'Framework' de desenvolvimento para aplicações baseadas na Internet, extremamente completa para desenvolvimento em PHP 5. É normalmente chamada de biblioteca de componentes que se podem usar de forma independente. Neste projeto é usado para a autenticação com a API @Google Apps.

### 2.9.4 - JQuery

A API JQuery é uma biblioteca concisa e de alto desempenho para Javascript que simplifica o manuseamento do HTML, eventos, animações e interações AJAX ou '*Asynchronous Javascript and XML*' para um desenvolvimento ágil em ambientes Web. É um elemento facilitador ao desenvolvimento em Javascript e é regido sob as licenças MIT e GPL, existindo inúmeros 'plugins' que auxiliam os programadores a desenvolver e partilhar funcionalidades.

A biblioteca poderá ser descarregada em <http://jquery.com> e sendo que poderá sê-lo em dois formatos:

- comprimido (permite a utilização com ficheiros de muito menor tamanho)
- não comprimido (preferível para depurações e ambientes de desenvolvimento pois permite uma leitura humanamente mais agradável)

Poderá, também, ser utilizado com recurso a uma CDN (Content Delivery Network) recomendada:

#### **Google Ajax API CDN (também com suporte SSL via HTTPS)**

- <http://ajax.googleapis.com/ajax/libs/jquery/1.7/jquery.min.js> (versão minimizada)
- <http://ajax.googleapis.com/ajax/libs/jquery/1.6.4/jquery.min.js> (versão minimizada)

### **Microsoft CDN (também com suporte SSL via HTTPS)**

- <http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.min.js>

### **jQuery CDN (via Media Temple)**

- <http://code.jquery.com/jquery-1.7.min.js> (versão minimizada)
- <http://code.jquery.com/jquery-1.7.js>

Este método agiliza o carregamento da página, uma vez que tratando-se de uma rede pública poderá estar já carregada pelo navegador ou 'cache'. É apoiada por grandes empresas que alojam cópias do JQuery numa rede CDN que são de utilização pública.

No entanto, a utilização recomendada é a tradicional, isto é, o descarregamento da biblioteca e alojamento independente garantindo assim que o ambiente não sofre alterações sem conhecimento do programador.

### **Como funciona**

A título de exemplo apresentarei um pequeno conjunto de instruções para melhor elucidar a utilização desta API. Assim, partindo do princípio que temos uma determinada página simples:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Demonstração</title>
  </head>
  <body>
    <a href="http://jquery.com/">jQuery</a>
    <script src="jquery.js"></script>
    <script>
//código
    </script>
  </body>
</html>
```

Deveremos alterar o atributo 'src' da Tag 'script' para o direcionar para a nossa

cópia do JQuery. Neste caso a biblioteca encontra-se na mesma diretoria que a nossa página de exemplo.

Executar código assim que for despoletado pelo navegador, que a página está completamente carregada, é normalmente feito através da instrução;

```
window.onload = function(){ alert("olá Mundo"); }
```

O javascript, no entanto, não é executado até que todos os elementos da página estejam carregados, incluindo-se as imagens, publicidade etc...

Para ultrapassar esta questão, o JQuery tem uma simples instrução que verifica o documento e espera até que fique pronto para ser manipulado:

```
$(document).ready(function(){  
    //código  
});
```

Dentro do evento 'ready' vamos adicionar um manipulador de eventos, 'handler', à hiperligação:

```
$(document).ready(function(){  
    $("a").click(function(event){  
        alert("Olá Mundo!");  
    });  
});
```

Ao gravar estas alterações e executando a página num navegador, constatamos que clicando na hiperligação da página, aparecerá uma janela de alerta antes de navegar para a página <http://jquery.com>.

Para cada clique e restantes eventos, é possível impedir o comportamento por omissão do navegador, neste caso navegar para <http://jquery.com>, através do manipulador de eventos, `event.preventDefault()`, substituindo o código anterior por:

```
$(document).ready(function(){  
    $("a").click(function(event){  
        alert("Podemos verificar que já não navegaremos para jquery.com");  
        event.preventDefault();  
    });  
});
```

Outra operação comum é adicionar ou remover classes de CSS a um determinado elemento HTML.

Por exemplo, adicionar a classe “teste” ao link poderá ser efetuado através da instrução:

```
$("#a").addClass("teste");
```

Todos os links da página ficarão sob influência da classe “teste”, qualquer que seja a definição desta classe.

Para fazer o inverso, remover uma classe poderemos usar a instrução:

```
$("#a").removeClass("teste");
```

Para trabalhar com animações e efeitos, que melhoram e muito a experiência do utilizador, poderemos fazer a seguinte alteração ao nosso código:

```
$("#a").click(function(event){  
    event.preventDefault();  
    $(this).hide("slow");  
});
```

Assim, após recarregar a página no navegador, verificaremos que, após clicar na hiperligação, este desaparecerá lentamente.

Referente a funções e ‘callbacks’ começarei por explicar que um ‘callback’ é uma função que é passada como argumento e é executada só após a função anfitriã estiver terminada.

A título de exemplo poderemos executar a seguinte instrução:

```
$.get('minhapagina.html', meuCallBack);
```

Ou, se tiver argumentos:

```
$.get('minhapagina .html', function(){  
    meuCallBack(param1, param2);  
});
```

Neste último caso a função ‘meuCallBack’ é executada pela função anónima, com os parâmetros “param1” e “param2” mas no âmbito exterior, estes são avaliados como ‘callback’ quando a função “\$.get” terminar de receber a “minhapagina.html”

Para mais informações sobre todos os métodos é disponibilizado o domínio <http://docs.jquery.com> onde se enumeram todos os métodos, argumentos e respetivos atributos.

Para auxiliar a utilização desta API, existe também um projeto que disponibiliza um conjunto de ‘plugins’, configuráveis no que diz respeito à experiência do utilizador, onde poderão ser descarregados, ou criados, temas que atuarão em conjunto com o JQuery

acelerando o processo de desenvolvimento no que diz respeito a controlos mais usados frequentemente e que poderão ser descarregados e utilizados ao abrigo das licenças MIT ou GPL. Estes são:

### **Núcleo – Ou ‘UI Core’**

Uma dependência necessária que contém todas as funções básicas e inicializadores

- 'Core' – O núcleo do jQuery UI, necessário para todas as interações e controlos.
- 'Widget' – O criador dos controlos sendo a base de todos eles.
- 'Mouse' – O controlo do rato. Uma classe base para todas as interações e controlos com utilização do rato.
- 'Position' – Um utilitário para o posicionamento de elementos e a sua relação com os restantes elementos.

### **Interacções – Ou ‘Interactions’**

Tem como objetivo adicionar comportamentos a qualquer elemento, sendo usado por muitos controlos.

- 'Draggable' – Torna qualquer elemento da página arrastável com o rato.
- 'Droppable' – Gera destinos onde se podem largar os elementos 'Draggable'
- 'Resizable' – Torna qualquer elemento da página redimensionável.
- 'Selectable' – Faz uma lista de elementos seleccionáveis com o rato ou por arrasto ou por clique.
- 'Sortable' – Torna uma lista de elementos ordenável.

### **Controlos – ‘Widgets’ ou controlos personalizáveis**

- 'Accordion' – Controlo de navegação do tipo acordeão
- 'Autocomplete' – Controlo que simula um campo de preenchimento automático com base num conjunto de caracteres inserido pelo utilizador
- 'Button' – Cria um botão compatível com o tema do jqueryui
- 'Dialog' – Abre um determinado controlo de HTML numa janela auxiliar
- 'Slider' - Um controle deslizante flexível com faixas e acessibilidade via teclado.
- 'Tabs' - Transforma um conjunto de elementos numa estrutura de separadores de conteúdo.

- 'Datepicker' – Um calendário que pode ser usado como entrada de um campo ou mostrado em linha.
- 'Progressbar' – Uma barra de progresso que pode ser usada para demonstrar um estado de um processamento.
- Efeitos – Ou 'Effects' permite a atribuição de animações e efeitos a elementos da página melhorando a experiência do utilizador. Aumenta o potencial dos efeitos nativos do JQuery e, não querendo fazer uma enumeração exaustiva dos mesmos, são eles: 'Blind', 'Bounce', 'Clip', 'Drop', 'Explode', 'Fade', 'Fold', 'Highlight', 'Pulsate', 'Scale', 'Shake', 'Slide', 'Transfer'.

### **Datatables – <http://www.datatables.net>**

O 'plugin'.datatables para JQuery é uma ferramenta altamente flexível, com ótima documentação, que permite tornar qualquer tabela HTML num controlo mais amigável do utilizador, e que permite uma parametrização extensa:

- Paginação de tamanho variável
- Filtragem de resultados
- Ordenação por colunas com deteção de tipo de dados
- Manuseamento inteligente da largura das colunas
- Disponibiliza dados provenientes de um largo espectro de fontes de dados
- Possibilidade de deslocamento dos dados na janela de exibição
- Compatível com qualquer tipo de mapa de caracteres
- Compatível com os temas do JQueryUI
- Completamente personalizável através de CSS
- Criação dinâmica de tabelas
- Carregamento automático via AJAX

### **Datatables Editable - <http://code.google.com/p/jquery-datatables-editable/>**

Este 'plugin' atua com dependência do 'plugin'.datatables, adicionando a este as seguintes funcionalidades:

- Adição de registos
- Seleção e remoção de registos
- Edição de células



Assim, adicionalmente a todas as funcionalidades do ‘plugin datatables’, é possível adicionar, seleccionar e remover registos tal como editar células da tabela, para estas funcionalidades adicionar necessita também, além do já referido ‘datatables’, os seguintes ‘plugins’:

- JQuery Editable – que permite a edição de qualquer elemento numa página HTML, disponível em <http://www.appelsiini.net/projects/jeditable>
- JQuery Validation – Permite a validação de campos de formulário, disponível em <http://bassistance.de/jquery-plugins/jquery-plugin-validation/>
- Todo o processamento é encapsulado neste ‘plugin’ e o único requisito é código de servidor que aceite as operações:
  - Adicionar registos
  - Atualizar dados
  - Remover registos

O código de servidor pode ser escrito em qualquer linguagem de programação desde que aceite os dados enviados.

#### **JQuery Ganttview - <https://github.com/thegrubbsian/jquery.ganttView>**

Este ‘plugin’ permite a criação de gráficos de Gantt em páginas HTML, sem a necessidade de imagens para os construir.

Suporta redimensionamento e arrastamento das barras de Gantt atualizando os dados através de envios, via AJAX, para código de servidor.

Como requisitos necessita, obviamente do JQuery, JQueryUI e de date.js, sendo este um utilitário disponibilizado em <http://www.datejs.com/>

### **3 - A Solução Ez-BUD**

Como anteriormente descrito por diversas vezes e como referido em todos os pontos de análise, nenhuma das soluções contempla as necessidades duma sociedade cada vez mais ágil que exige mobilidade e segurança da informação. Com vista a minimizar este problema, decidiu-se desenvolver uma aplicação multi-plataforma com recurso às mais recentes tecnologias a qual foi batizada de Ez-Bud. A solução Ez-Bud, pretende apostar e distinguir-se face às demais aplicações tendo as seguintes funcionalidades, numa primeira fase, e dentro do âmbito desta dissertação.

- Cálculo de Preços e Fichas de Rendimento
- Elaboração de Cadernos de Encargos e Elaboração de Orçamentos
- Cronogramas financeiros e Mapas de Trabalho

As aplicações analisadas anteriormente, são, claro, aplicações muito completas com as quais o Ez-Bud não poderá concorrer no seu todo, nesta primeira fase, que se centra apenas no tema orçamentação e tratamento de preços, sendo que, numa segunda fase, serão implementados os módulos de Faturação e Controlo de custos, permitindo assim acompanhar a obra desde a sua origem até ao seu término e onde serão também contempladas as respetivas trocas de informação com os mais conhecidos ERP na vertente de serviço ‘Premium’.

A nível de integração o Ez-Bud deverá contemplar uma opção de SSO (Single Sign-ON) permitindo assim uma maior interação com plataformas já existentes, dispensando o registo de utilizadores e otimizando o conceito ‘Nuvem’ envolvendo assim todos as condicionantes que a sociedade atual exige nomeadamente mobilidade e redução de custos de licenciamento e manutenção técnica entre outros Requisitos Não Funcionais.

#### **3.1 - ‘Stakeholders’**

Por definição, um ‘Stakeholder’ é uma pessoa, grupo ou organização que tenha participação direta ou indireta numa organização, porque pode afetar ou ser afetado pelas ações da organização, objetivos e políticas. Em aplicações informáticas o mesmo se passa uma vez que uma aplicação informática interage ou modifica processos que por sua vez tem participação de diversos intervenientes.

No caso do Ez-Bud são todas as pessoas ou grupos que têm participação num orçamento, assim sendo temos:

- Cliente
- Fornecedores
- Orçamentista ou qualquer outra entidade que tenha a responsabilidade de calcular os preços de custo e de venda de determinado levantamento de preços pedido por um cliente

Ainda que nesta primeira fase deste projeto não haja, já, a participação de todos os intervenientes, é pertinente considerá-los de imediato. Refiro-me aos fornecedores que só numa segunda fase terão participação neste projeto.

### **3.2 - Levantamento de requisitos**

O levantamento de requisitos é a fase mais crítica no desenvolvimento de aplicações informáticas, sendo que o sucesso depende do detalhe que foi dado nesta fase. O levantamento de requisitos pode ser dividido em requisitos funcionais e não funcionais. Geralmente, os requisitos funcionais são expressos na forma do que o 'sistema deve fazer', enquanto os requisitos não funcionais são o que o "sistema deve ser".

#### **3.2.1 - Requisitos Não funcionais**

Requisitos não-funcionais, também conhecido como requisitos de qualidade, são os requisitos relacionados com o desempenho, usabilidade, fiabilidade, segurança, disponibilidade, manutenibilidade e tecnologias envolvidas. Em geral, requisitos não-funcionais podem constituir restrições aos requisitos funcionais uma vez que estes são características mínimas de um software de qualidade.

##### **3.2.1.1 - Licenciamento**

O Ez-Bud deverá ter soluções que permitam a utilização sem licenciamento.

##### **3.2.1.2 - Requisitos de Implementação**

- Interfaces de comunicação
  - Protocolo HTTP
- Interfaces de Hardware (mínima configuração)
  - Processador x86
  - 512Mb RAM
  - 500Mb HDD
- Interfaces de Software

- Windows 2008 Server / Linux 2.6.x ou superior
- MySQL 5.5 ou superior
- PHP 5.3 ou superior
- Apache 2.x ou superior
- Conformidade HTML5
- Portabilidade
  - Total entre Windows, Linux, Mac e Solaris com as restrições descritas em interfaces de software.

### 3.2.1.3 - Requisitos de Fiabilidade

O Serviço deverá ter uma disponibilidade de 99,999%. Este número é conhecido por 'cinco noves' ou 'five nines' e implica um acordo de nível de serviço no que diz respeito à disponibilidade do serviço. Para melhor entender poderemos verificar a seguinte tabela com os tempos de inatividade:

Disponibilidade %	Inatividade por ano	Inatividade por mês*	Inatividade por semana
99,00%	3.65 dias	7.20 horas	1.68 horas
99.5%	1.83 dias	3.60 horas	50.4 minutos
99.8%	17.52 horas	86.23 minutos	20.16 minutos
99.9% (três noves)	8.76 horas	43.2 minutos	10.1 minutos
99.95%	4.38 horas	21.56 minutos	5.04 minutos
99.99% (quatro noves)	52.56 minutos	4.32 minutos	1.01 minutos
99.999% (cinco noves)	5.26 minutos	25.9 segundos	6.05 segundos
99.9999% (seis noves)	31.5 segundos	2.59 segundos	0.605 segundos
* cálculo efetuado tendo como base meses de 30 dias			

*Tabela 4: Disponibilidade de serviço*

### 3.2.1.4 - Requisitos de Segurança

- Comunicações cifradas a 256bits (SSL)
- Autenticação via OpenID - O Ez-Bud contemplará todas as plataformas que tiverem um sistema de autenticação baseado em OpenID. Assim, quem tiver um 'login' já existente poderá com facilidade começar a usar a plataforma bastando para isso usar o seu 'login' OpenID a partir das @Google Apps.

### **3.3 - Requisitos Funcionais**

Em engenharia de software, um requisito funcional define uma função de um sistema de software ou do seu componente. Uma função é descrita como um conjunto de entradas, o comportamento, e saídas.

Requisitos funcionais podem ser cálculos, detalhes técnicos, a manipulação e processamento de dados e outras funcionalidades específicas que definem o que um sistema é suposto realizar. Requisitos comportamentais descrevendo todos os casos em que o sistema utiliza os requisitos funcionais são capturados em casos de uso.

Os requisitos funcionais são suportados pelos requisitos não-funcionais, que impõem restrições quanto ao projeto ou implementação.

#### **3.3.1 - Preços compostos**

Esta componente permitirá a estruturação dos recursos em quatro categorias – pessoal, equipamentos, materiais e sub-empreitadas. Futuramente a atualização de preços poderá ser efetuada individualmente, através de um fator, com base nos custos registados em obra. Permitirá também a estruturação de fichas de rendimento, com a composição qualitativa , que componentes, e quantitativa, que rendimentos de trabalhos. Os componentes poderão ser recursos ou outras fichas de rendimento, associados aos respetivos preços.

Este é, no fundo, o módulo base de suporte à gestão de preços compostos nos orçamentos, pois poderá ser a partir das fichas de rendimento que o orçamento de custo é elaborado.

#### **3.3.2 - Orçamentos**

Estruturação livre de mapas de trabalho, em itens e grupos, para elaboração de orçamento de custo e orçamento de venda.

O preço composto resulta das fichas de rendimento com os respetivos recursos, enquanto os restantes poderão ser introduzidos diretamente no orçamento.

Várias formas para manipulação do valor do orçamento serão possíveis para apresentar ao cliente, nomeadamente, alterando o valor total geral e totais dos grupo. Será também possível obter o valor de venda a partir do orçamento-custo e um fator k, refletindo-se no total do orçamento. Futuramente será ainda possível enviar diretamente itens do orçamento para ‘convidados’ (ex. Sub-empregados) que terão acesso apenas a essa parte

do orçamento para inclusão dos seus preços sendo assim uma plataforma colaborativa para a elaboração de orçamentos.

### **3.3.3 - Cronogramas**

Este módulo será responsável pela elaboração dos programas de trabalho, Permitirá obter, para cada programa de trabalhos, o cronograma financeiro, plano de trabalhos, planos de recursos e futuramente o mapa de carga de pessoal e mapa de carga dos equipamentos.

### 3.4 - Arquitetura da solução

Em termos aplicacionais, o Ez-Bud está estruturado segundo o modelo de três níveis, em que se podem identificar:

**Nível de Dados** – Componentes de acesso a dados responsáveis por manter a integridade da base de dados. Interagem diretamente com a base de dados a pedido dos componentes do nível lógico.

**Nível Lógico** – Componentes que implementam as regras de negócio, verificam a sua validade, efetuam todos os cálculos necessários para a aplicação. Interagem diretamente com os componentes de acesso a dados a pedido da interface.

**Nível de Apresentação** – Interface com o utilizador final, verificação da validade dos dados introduzidos pelo mesmo, apresentação de resultados ou mesmo erros se for caso disso. A interface interage diretamente com os componentes do nível lógico.

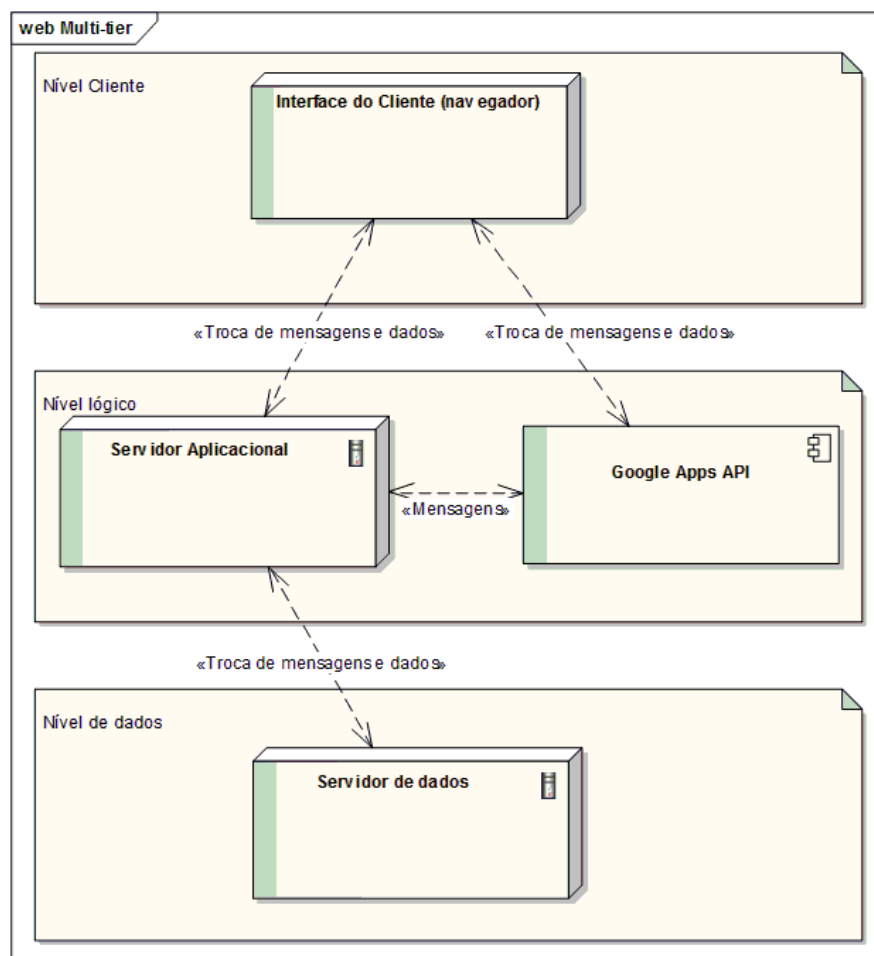


Figura 4: Arquitetura 'Three-tier'

Os níveis descritos têm uma localização física e recorrem a um número de máquinas que pode variar de acordo com as necessidades.

Neste capítulo será descrita a arquitetura da solução começando pelo estudo da interface e a funcionalidade multi-idioma que permitirá o acesso a qualquer utilizador em qualquer parte do mundo. Seguidamente serão discriminadas todas as tecnologias que serviram de base ao desenvolvimento da solução e, posteriormente, os níveis lógico e de dados.

### 3.4.1 - Estudo da interface

O estudo da interface partiu de uma visão minimalista com vista a qualquer utilizador não ter muitos obstáculos visuais.

Foi feito um levantamento de tarefas, da perspetiva do utilizador final, vulgarmente conhecido como 'histórias', pelo método de desenvolvimento de projetos 'Scrum'<sup>11</sup> (Cohn, 2005), que teve como principal preocupação que o utilizador chegue a qualquer funcionalidade em menos de 3 cliques ou seleções, dando origem à seguinte lista:

- Deve ter a possibilidade de manter uma base de dados de preços simples.
- Deve ter a possibilidade de criar e editar orçamentos.
- Deve ser possível a categorização de orçamentos.
- Deve ter a possibilidade de criar e editar mapas de trabalho.

Claro está que esta lista parte do princípio que o utilizador já se encontra a aplicação e não dentro de qualquer área das aplicações Google.

Relativamente à resolução foi implementada uma solução que permite magnificação automática em dispositivos móveis que o permitam através da seguinte diretiva:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

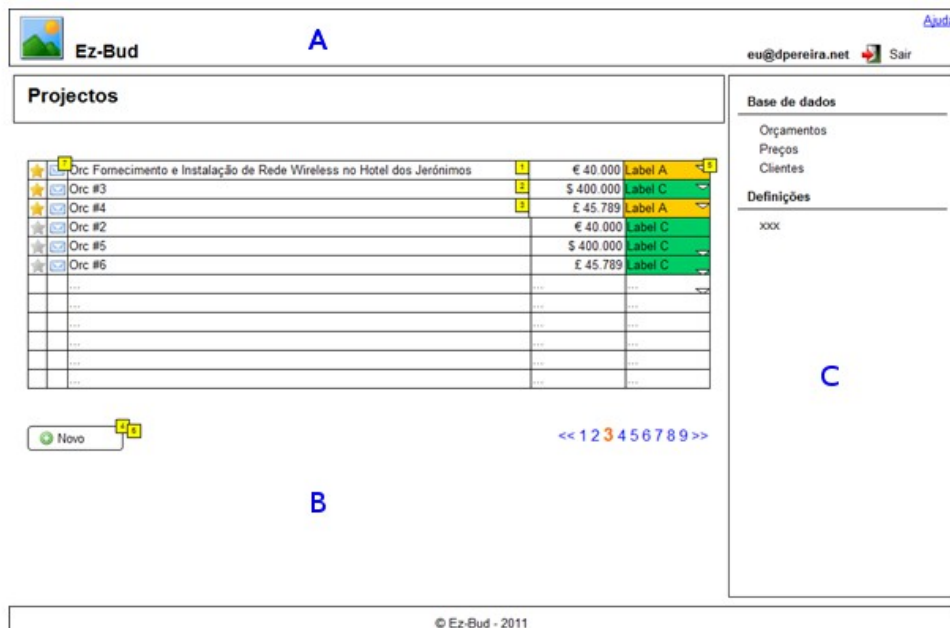
Para dispositivos que não tenham esta funcionalidade a resolução mínima é de 1024x600 pixels.

---

<sup>11</sup> O Scrum é um processo de desenvolvimento gestão de projetos e desenvolvimento ágil de aplicações informáticas



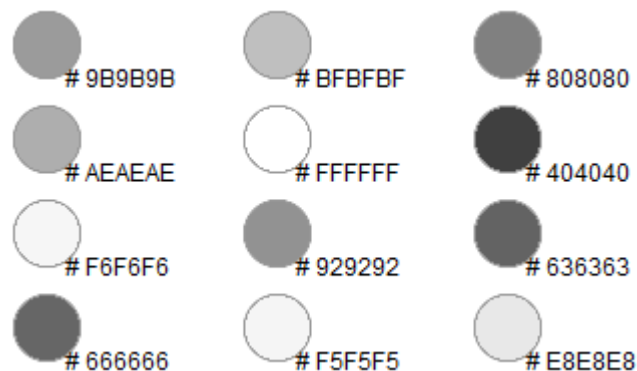
O estudo resultante deu origem ao seguinte protótipo:



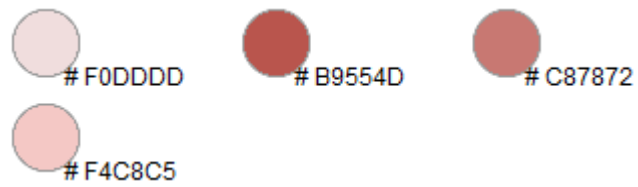
- A) Cabeçalho  
B) Conteúdo principal  
C) Auxiliar de navegação

Assim, com base nos conceitos verificados no protótipo foi efetuado um estudo de cores a aplicar cujo requisito de escolha foi usabilidade em vários dispositivos.

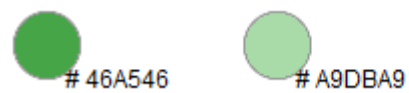
## Cinzentos



## Vermelhos



## Verdes



## Azuis



Aplicando este estudo de cores conceberam-se os seguintes ecrãs:

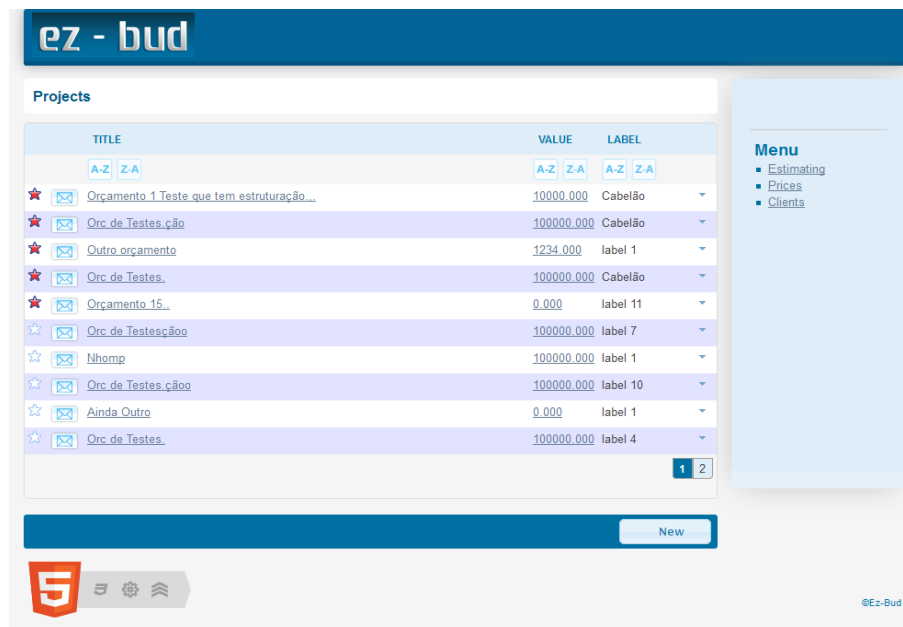


Figura 5: Ecrã principal do módulo de orçamentos

Na Figura 3 está o ecrã principal do módulo de orçamentos que consiste na lista de orçamentos já efetuados para que o utilizador possa seleccionar um para trabalhar ou efetuar um novo orçamento. O utilizador tem, também, a possibilidade de categorizar os orçamentos e visualizar o valor dos mesmos.

Na Figura 4, o utilizador visualiza o separador que diz respeito ao detalhe de um orçamento podendo inserir ou alterar os seguintes dados:

- Cliente
- Nome do orçamento
- Valor
- Data de submissão
- Data de Adjudicação
- Data de Consignação
- Moeda
- Notas

**ez - bud**

Projects

Estimate Structure Prices Schedule

Estimate Details

Client: David Cliente Pereira 2

Name: Orçamento 1  
Teste que tem estruturação...

Value: 10000.000

Submission Date: 2011-10-12

Adjudication Date: 2011-07-25

Consignment Date: 2011-07-26

Currency: \$

Notes: Notas e tal com outras coisas

New Reset Save

Menu

- Estimating
- Prices
- Clients

Print

- Estimate (Sale)
- Estimate (Cost)
- Schedule

Export

- MPX format

5

©Ez-Bud

Figura 6: Ecrã de edição de orçamentos

No separador seguinte, mostrado na Figura 5, encontra-se a estrutura do orçamento, que por omissão mostra sempre o primeiro nível, por exemplo capítulos, os quais podem ser abertos pelo utilizador clicando no símbolo '+’.

**ez - bud**

Projects

Estimate Structure Prices Schedule

Code	Description	Un	Qt	Price	Cost	Sale
4	Com 2 preços simples		1.500	12.000	18.000	28.800
14	14				42.000	42.000
141	Preço composto de 3º nível				14.000	14.000
1411	Aqui...	un	1.000	14.000	14.000	14.000
142	Com preço composto		2.000	14.000	28.000	28.000

Menu

- Estimating
- Prices
- Clients

Print

- Estimate (Sale)
- Estimate (Cost)
- Schedule

Export

- MPX format

5

©Ez-Bud

Figura 7: Edição da estrutura do orçamento

O detalhe de cada um dos itens do orçamento é mostrado na Figura 6, onde o utilizador pode inscrever todos os dados do respetivo item:

- Código
- Descrição
- Unidade
- Quantidade
- Preço

Alternativamente o preço pode ser composto por fichas de rendimento pelo que, neste caso, terá que seleccionar uma ficha de rendimento através da inserção do código, ou nome do recurso que quer utilizar, na caixa pesquisar.

The screenshot displays the 'EZ-BUD' application window for editing an item. The 'Description' field contains the text 'Com preço composto'. The right-hand side of the window features several input fields: 'Code' (142), 'Un' (Unit), 'Qt' (Quantity: 2.000), 'Unit Price' (14.000), 'Margin' (1.000), 'Total Cost' (28.000), and 'Total Sale' (28.000). Below these fields is a search bar with the placeholder text 'Type part of the code or description here'. A table below the search bar shows one entry: '50001 Lavadora.' with 'ml' as the unit, '14.000' as the price, '1.000' as the quantity, and '14.000' as the total. At the bottom of the window are 'Delete', 'Reset', and 'Save' buttons.

Code	Name	Unity	Price	Quantity	Total
50001	Lavadora.	ml	14.000	1.000	14.000

Figura 8: Edição de um item de um orçamento

Na Figura 7, mostra-se o ecrã para a definição do plano de trabalhos. Fazendo duplo clique em cima do item a definir a data de início e fim ou, caso já exista, basta arrastar as extremidades para aumentar ou diminuir o tempo de execução do item.

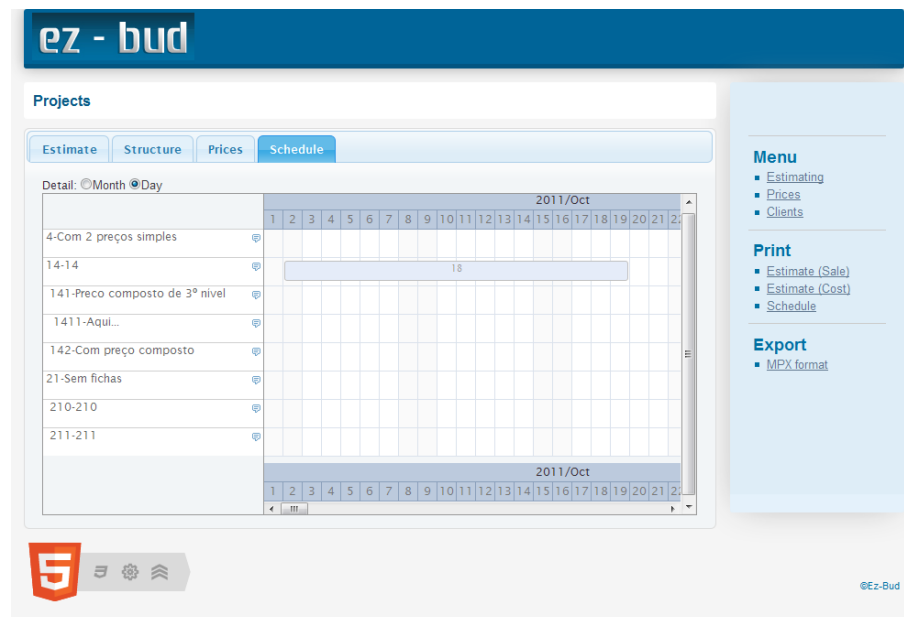


Figura 9: Edição de mapas de trabalho

É também possível, em todo o orçamento imprimir o orçamento a preço de custo ou venda assim como o plano de trabalhos ou cronograma.

Na Figura 8, mostra-se o ecrã para edição de clientes. Como qualquer outra tabela em toda a aplicação o clique fará a seleção ou visualização do detalhe, ao passo que o duplo clique indicará que se trata de uma edição do registo diretamente na tabela.

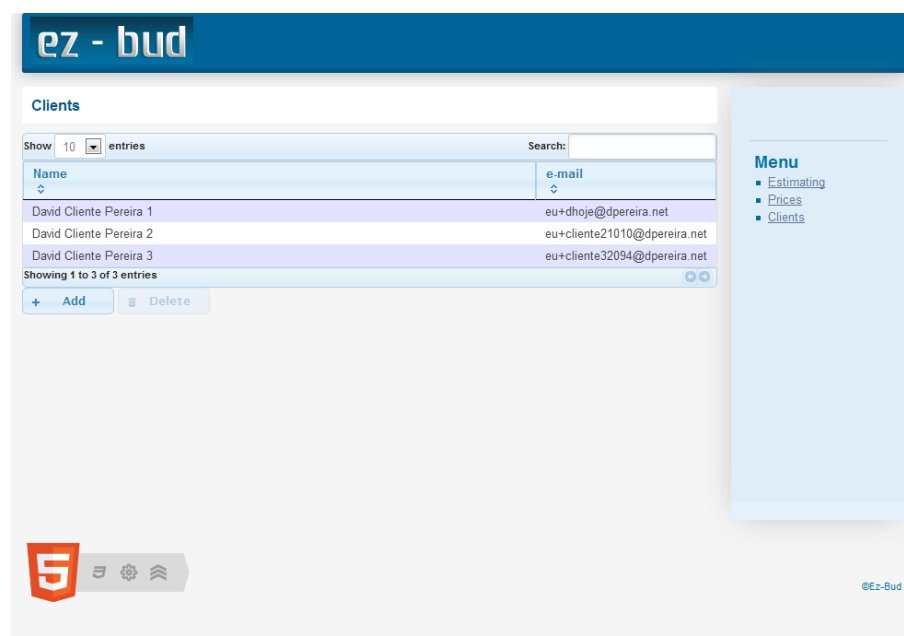


Figura 10: Edição de clientes

Na Figura 9, podemos visualizar o módulo de preços compostos com os seus separadores para cada tipo de recurso, funcionando todos da mesma forma, isto é, duplo clique para editar os valores ou um único clique para seleccionar o recurso ou componente.

**ez - bud**

**Resources**

Labor Supplies Equipments Subcontractors **Composite prices**

Show 10 entries Search:

Code	Name	Unity	Price
50001	Lavadora	ml	7.950
50002	Doobro nham	un	16.400
50003	gasp 3	un	1.000

Showing 1 to 3 of 3 entries (filtered from 7 total entries)

+ Add Delete

Type part of the code or description here

Show 10 entries Search:

Code	Name	Unity	Price	Quantity	Total
3	EQP	ml	1.650	3.000	4.950
1	pedreiro	Hr	1.000	3.000	3.000

Showing 1 to 2 of 2 entries

Delete

**Menu**

- Estimating
- Prices**
- Clients

5

©Ez-Bud

Figura 11: Edição de preços - separador de preços compostos

### **3.4.2 - Multi-idioma**

A plataforma Ez-Bud está desenhada para poder ser disponibilizada em qualquer idioma, sendo que, no âmbito desta dissertação estará implementada apenas em inglês e português. Esta necessidade advém do facto de se destinar ao mercado global onde qualquer utilizador de ®Google Apps pode subscrever e utilizar a aplicação. A estrutura de dados compreende o facto de poder haver tantos idiomas quantos necessários.

A possibilidade de ter os dados inseridos pelo utilizador em mais que um idioma foi relegada por representar um grande obstáculo à interface do utilizador, no entanto, por estar em utf-8 qualquer idioma pode utilizar a aplicação uma vez que aceita todos os tipos de caracteres.



### 3.4.3 - Arquitetura de dados

A arquitetura de dados foi pensada para permitir futuros desenvolvimentos, uma vez que o Ez-Bud será uma plataforma mais extensa não se limitando ao tema da orçamentação. Assim para cumprir com os requisitos funcionais e tendo sempre em consideração o desempenho foi criada a estrutura da Figura 11 para base de dados MySQL com tabelas do tipo @InnoDB.

InnoDB é o motor de armazenamento padrão para o MySQL a partir da versão 5.5. Fornece as características padrão ‘ACID-compliant’<sup>12</sup> de transação, juntamente com o suporte de chave estrangeira (integridade referencial declarativa).

Esta arquitetura permite que o utilizador possa ter a sua base de dados de preços constantemente atualizada e que possa alterar os seus preços nos orçamentos efetuados sem alterar a base de dados de preços central. A funcionalidade multilingue da aplicação também está espelhada neste esquema, uma vez que o público alvo da aplicação Ez-Bud é global, não se limitando a um país ou região. Deste modo toda a estrutura é compatível com o mapa de caracteres UTF-8, que permite uma maior liberdade de utilização de caracteres de todos os idiomas.

---

<sup>12</sup> ACID (atomicity, consistency, isolation, durability) é um conjunto de propriedades que garante que as transações na Base de Dados são processadas confiavelmente.

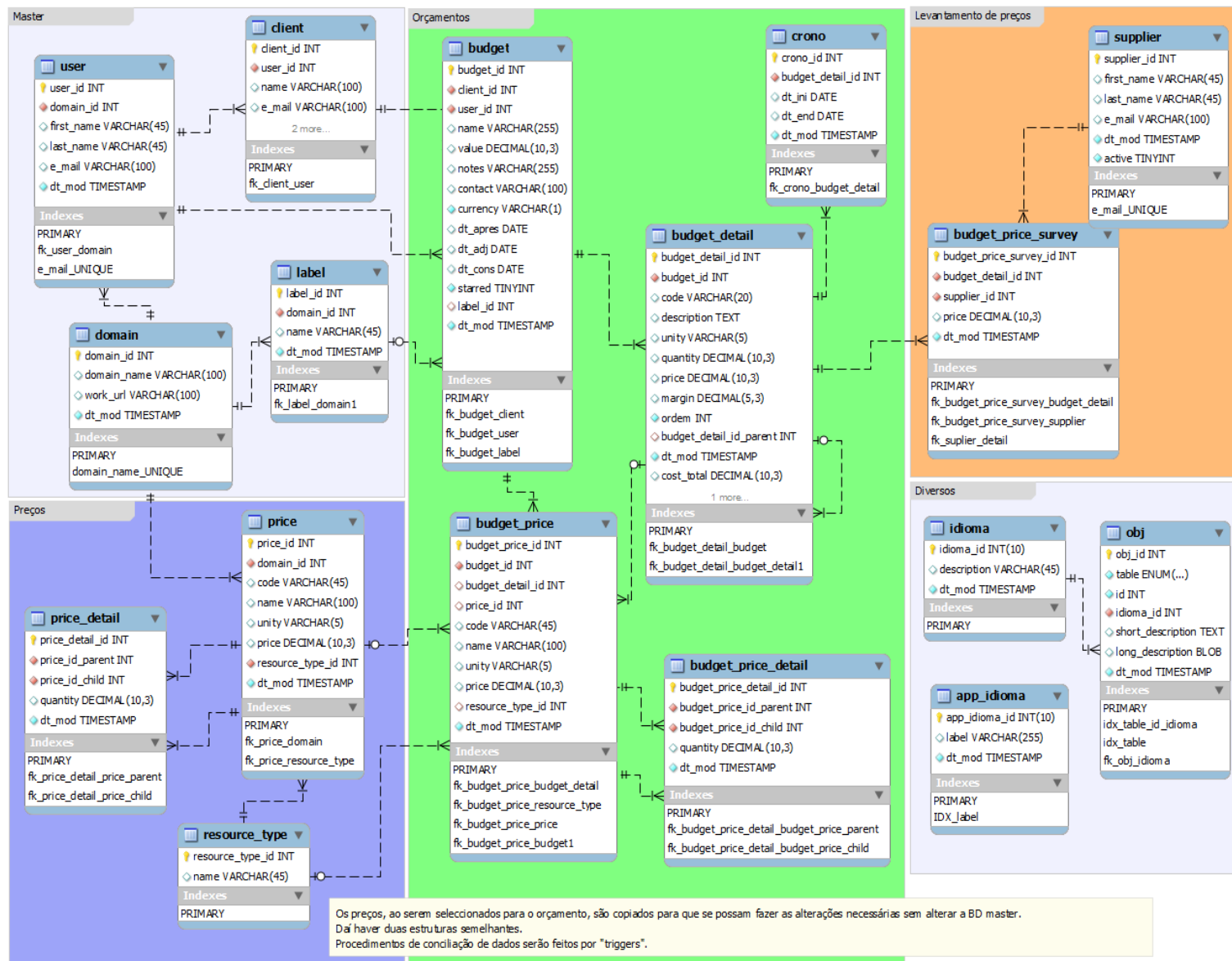


Figura 12: Modelo EER - Entity Relationship Model ou Modelo de Relacionamentos de Entidades

### 3.4.3.1 - Entidades principais

De seguida serão descritas as principais Entidades presentes na Figura 11.

**Domain** – Esta entidade rege quais os domínios que estão devidamente autorizados a utilizar a aplicação. É atualizada sempre que uma nova entidade instala a aplicação Ez-BUD.

**User** – Após a instalação, o administrador do domínio que está a instalar a aplicação terá a possibilidade de associar utilizadores do domínio à aplicação. É nesta entidade que ficam registados os utilizadores autorizados à utilização da aplicação.

**Resource\_type** – Define os tipos de componentes. É uma entidade que só sofrerá atualizações se o programador decidir criar um novo tipo de componente. De início esta entidade apenas tem os registos:

1. Mão-de-Obra
2. Materiais
3. Equipamentos
4. Sub-empreitadas
5. Fichas de rendimento.

**Price** – Esta entidade define os preços da base de dados central para cada um dos domínios autorizados.

**Budget** – O orçamento e respetivos dados são definidos nesta entidade.

**Budget\_detail** – A estrutura do orçamento é definida por esta entidade sendo que cada registo tem a informação do seu registo raiz.

**Budget\_price** - Esta entidade define os preços de um orçamento para cada um dos domínios autorizados

**Crono** – Define as temporizações referentes a cada um dos registos da entidade budget\_detail.

**Idioma** – Esta entidade define quais os idiomas ativos na aplicação.

**App\_idioma** – A relação das 'labels' devidamente usadas por toda a aplicação.

**Obj** – Designação para cada um dos idiomas dos registos criados na entidade 'app\_idioma'.

#### 3.4.4 - Arquitetura lógica

Como podemos analisar na Figura 12, a plataforma está limitada a entradas apenas por uma página com recurso ao módulo 'mod\_rewrite' do servidor de páginas Apache. Este permite a reescrita dos URL que acedem à aplicação mesmo quando é pedida uma página que não existe.

Assim quando se acede a uma página 'http://www.ez-bud.com/naoexiste' o utilizador é redirecionado, de forma invisível para 'http://www.ez-bud.com/index.php?do=Dashboard'. Por sua vez a página index.php faz todo o tratamento necessário para mostrar o painel de controlo da aplicação.

De igual forma são geridos as reais 'portas de entrada':

- /Prices
- /Clients
- /Estimate

Relativamente à estrutura de ficheiros que compõem a aplicação, podemos descrever a seguinte hierarquia:

- css – CSS utilizadas na aplicação
- img – Todas as imagens da aplicação
- inc – Todo o código lógico
- js – Ficheiros de javascript
- SSO – Ficheiros e 'Frameworks' utilizadas na autenticação com Google Apps
- index.php
- favicon.ico – Icon que identifica o URL
- .htaccess

### 3.4.4.1 - Comunicação e mensagens

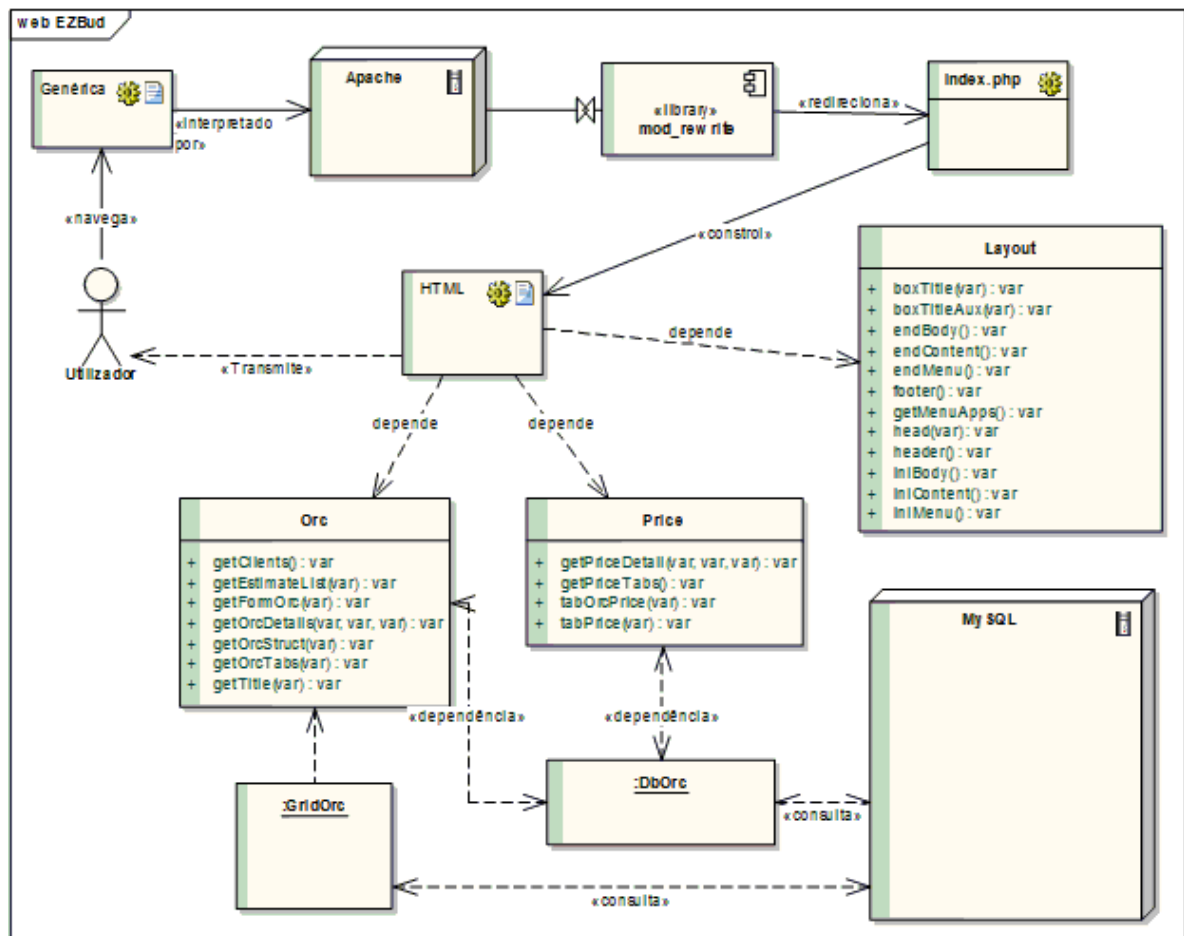


Figura 13: Fluxo de mensagens - Generalidade

O comportamento de redirecionamento pelo 'mod\_rewrite' do servidor de páginas Apache, resulta da colocação de um ficheiro '.htaccess' na raiz da aplicação:

```

AddDefaultCharset UTF-8
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteBase /
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteCond %{REQUEST_URI} !^/index.php
    RewriteRule (.*) index.php?do=$1
</IfModule>
    
```

Através de pedidos parcelares ao servidor, a experiência do utilizador é enriquecida uma vez que não existe refrescamento das páginas mas sim pedidos síncronos ou assíncronos, vulgo AJAX, de partes da página.

Este funcionamento não é apenas uma questão de experiência do utilizador mas também de desempenho uma vez que os pedidos são efetuados e só são pedidas ou escritas as partes que são necessárias para o utilizador poder desempenhar a sua função. Dentro da aplicação todos os pedidos são efetuados por método 'POST', com recurso a funções de Javascript, sendo o seu diagrama descrito na Figura 13:

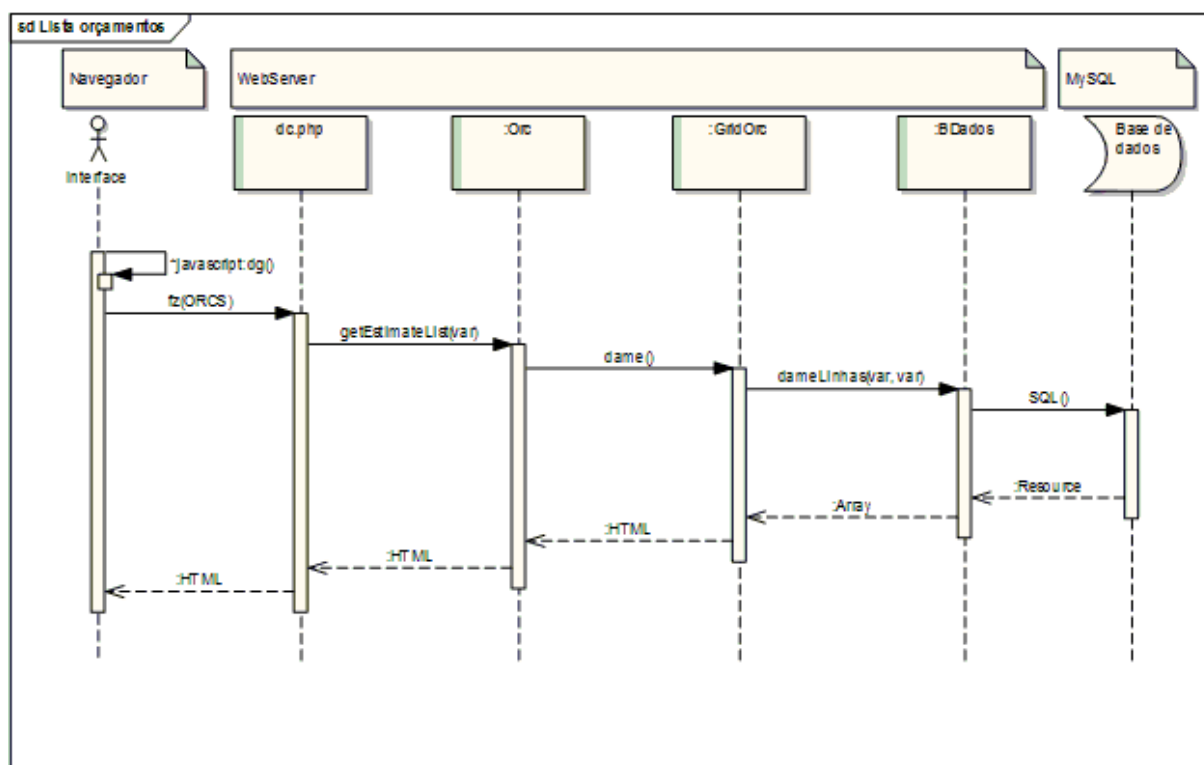


Figura 14: Fluxo de mensagens – ‘Use Case’ lista de orçamentos

Do lado do servidor foram desenhadas classes de programação para melhor organização do código e escalabilidade. O diagrama de classes é o demonstrado na Figura 14.

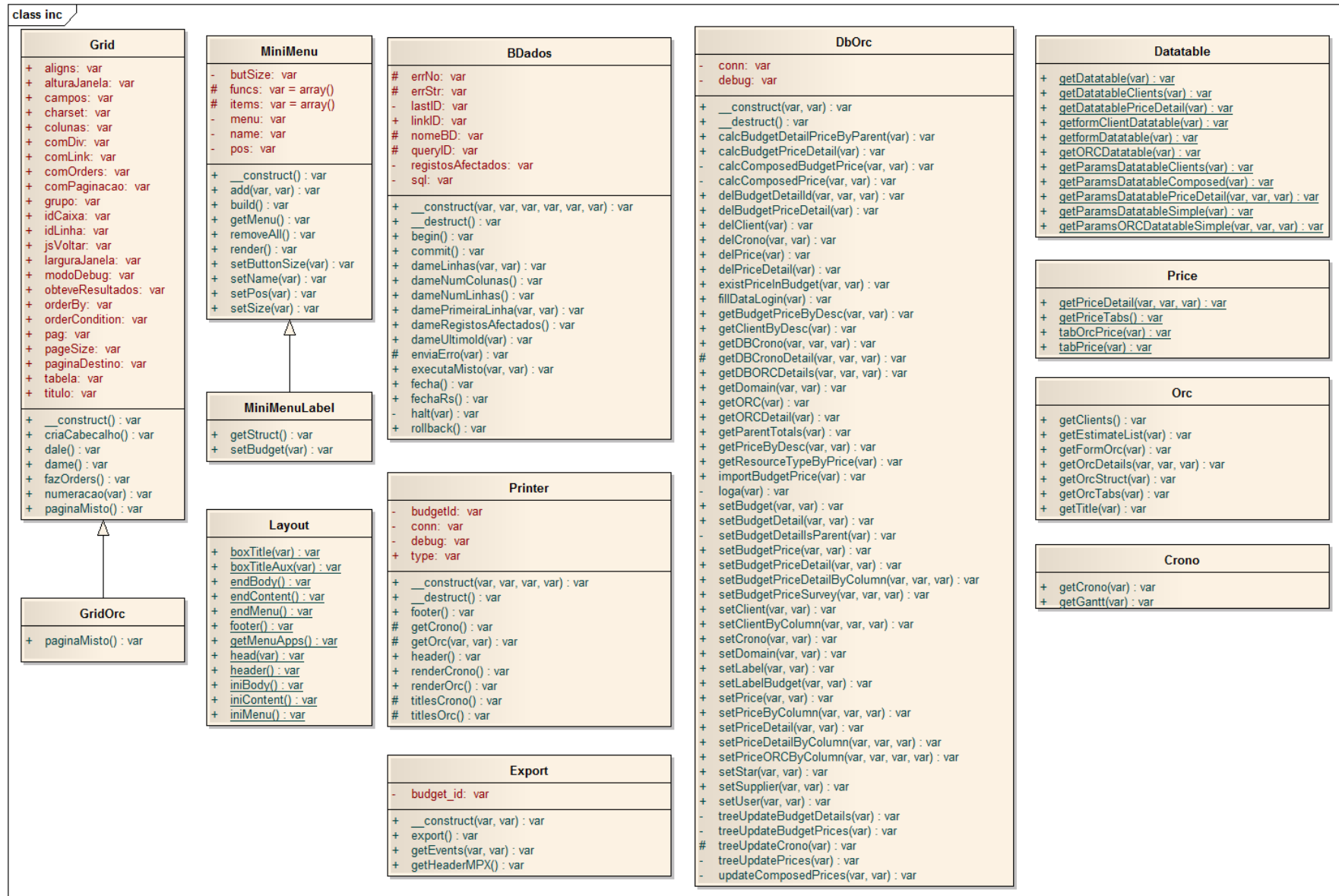


Figura 15: Diagrama lógico

## 4 - Metodologia

O presente capítulo tem como principal objetivo ilustrar qual a metodologia seguida ao longo do projeto, as fases por que passou e quais as consequências da adoção de tal metodologia de trabalho. Para uma melhor compreensão dividiu-se este capítulo em três secções distintas:

- Planeamento do projeto.
- Análise de riscos do projeto.
- Fases do projeto e metodologia utilizada em cada uma delas.

### 4.1 - Planeamento

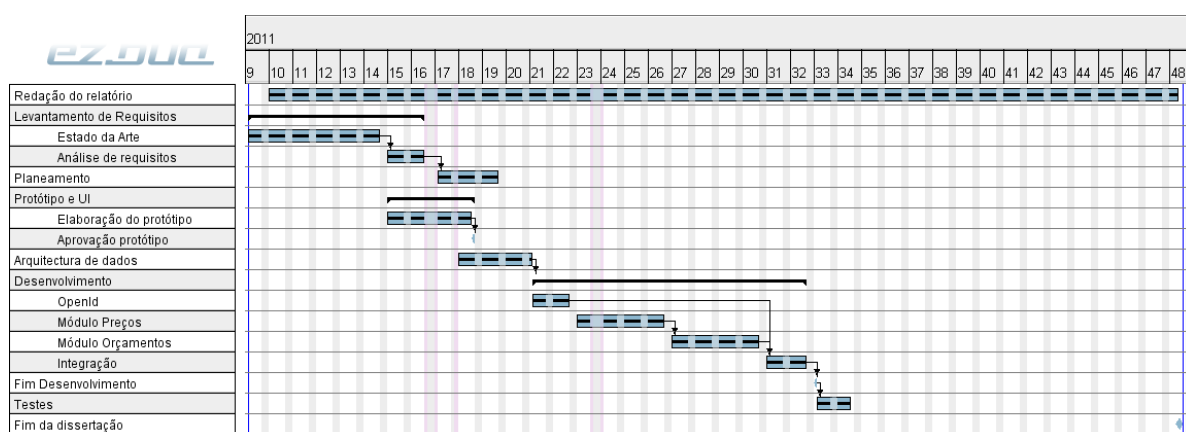


Figura 16: Plano de trabalhos (semanas)

### 4.2 - Análise de riscos

A análise de riscos consistiu na verificação da necessidade de construir esta plataforma modular, estando consciente da imensa oferta de soluções semelhantes. No que respeita à dimensão do projeto e recursos envolvidos, facilmente se depreende que é um projeto de grande dimensão, que obriga à gestão de uma grande quantidade de informação, assim como a devida estruturação do código para que se consiga interagir com as várias plataformas e componentes envolvidos, que se caracterizam pela sua heterogeneidade.



### **4.3 - Fases do projeto**

Sem grande espanto, chegamos à conclusão que as fases de análise e desenho foram as mais detalhadas, embora o gráfico não o represente, teremos que ter em conta que das 6 semanas de desenvolvimento só foram na realidade 48 horas (8 horas x 6 Semanas) ao passo que a análise e desenho foram 88 horas (8 horas x 11 dias) totalizando um total de 136 horas de desenvolvimento não incluindo a respetiva documentação que este relatório faz parte.

#### **4.3.1 - Requisitos, Planeamento e Protótipo**

Esta fase foi interessante na medida em que, criar um projeto novo, requer alguma investigação o que é sempre enriquecedor. Javascript, PHP e MYSQL são um universo de oportunidades e funcionalidades ao nosso dispor.

#### **4.3.2 - Desenho e Construção das Bases de Dados**

O desenho da base de dados centrou a sua preocupação na funcionalidade multilingue, pesando sempre o impacto que essa funcionalidade pode provocar na plataforma. Penso ter sido conseguida a funcionalidade sem que, nos testes, se tenha notado perda de desempenho.

#### **4.3.3 - Desenvolvimento e Programação**

O desenvolvimento decorreu de acordo com o planeado, recorrendo a metodologia ágil isto é, com pequenos ciclos de desenvolvimento (Cohn, Mike, 2005). O acesso à documentação das diversas API envolvidas por vezes levantou alguns problemas que tiveram que ser resolvidos com recurso a alguma criatividade pois estas não permitiam fazer algumas funcionalidades de raiz.

#### **4.3.4 - Testes à aplicação**

Após geração do código, há a necessidade de efetuar testes de aceitação (Pressman, Roger S., 2001), garantindo que os requisitos funcionais estão implementados e que não existem erros. Normalmente desprezada no que toca ao planeamento foi, mais uma vez, responsável pelo acréscimo de trabalho não previsto, na reta final da aplicação.

## 5 - Ferramentas utilizadas

Neste capítulo descrevem-se todas as ferramentas utilizadas, tanto no desenvolvimento da aplicação, como da própria dissertação. Mais uma vez foi dada primazia a aplicações de código aberto ou de utilização livre.

### 5.1 - WAMP

O WampServer é um ambiente de desenvolvimento para o sistema operativo @Windows permitindo desenvolver aplicações com suporte a Apache, PHP e Sistema de Gestão de Base de Dados MySQL. Também é fornecido a aplicação PHPMyAdmin para gerir as bases de dados. Tem uma instalação muito fácil e intuitiva sendo possível também configurações mais complexas através da gestão dos ficheiros de configuração. Através de um ícone é possível aceder a qualquer configuração ou serviço instalado.

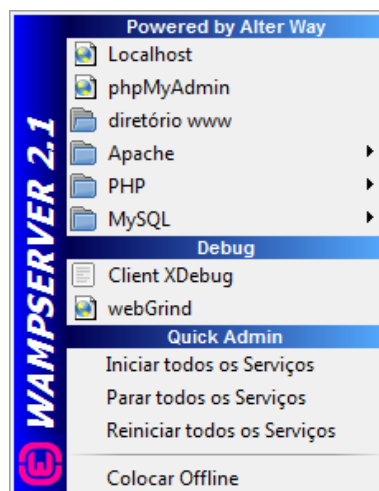


Figura 17: Menu de controlo do WampServer

É um ótimo pacote de aplicações para desenvolver uma vez que engloba todas as tecnologias envolvidas nesta dissertação.

O sistema de licenciamento é GPL Versão 2 e o código está disponível em <http://sourceforge.net/projects/wampserver/>

## 5.2 - Axure RP

A única aplicação que não é gratuita mas as suas vantagens na elaboração de um protótipo funcional são imensas. Permite fazer um esboço da interface do utilizador, com interações, auxiliando muito o processo de definição e interações com o utilizador.

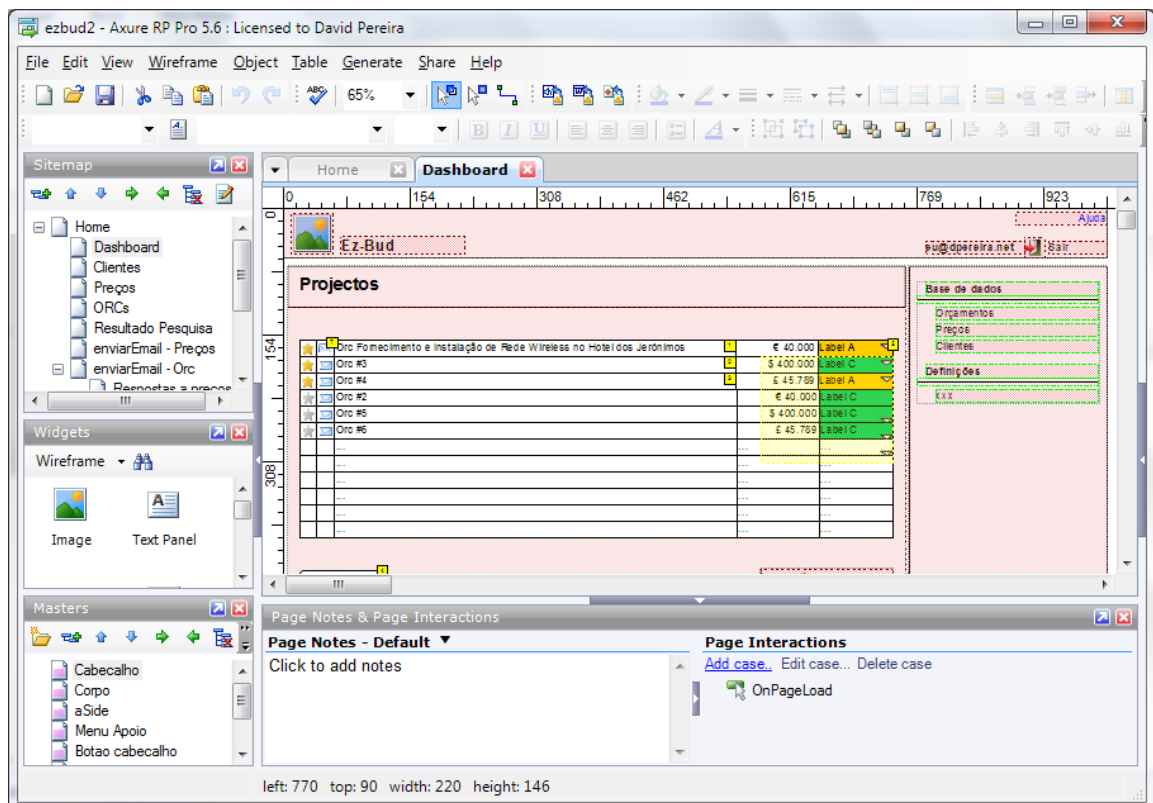


Figura 18: Ecrã da aplicação Axure RP

O sistema de licenciamento é proprietário e a versão da avaliação está disponível em <http://www.axure.com/download>

### 5.3 - Netbeans

O IDE Netbeans<sup>13</sup>, ver Figura 19, é um ambiente de desenvolvimento integrado disponível para ®Windows, ®Mac, Linux e Solaris e consiste num projeto de código aberto que possibilita aos programadores a criação de aplicações, sejam elas tradicionais, Web ou móveis. As linguagens de programação permitidas são Java, PHP, Javascript, Groovy and Grails e C/C++. Este projeto é suportado por uma comunidade de programadores e oferece uma documentação extensa assim como extensões de terceiros que aumentam a sua versatilidade (Ex. Modelação UML).

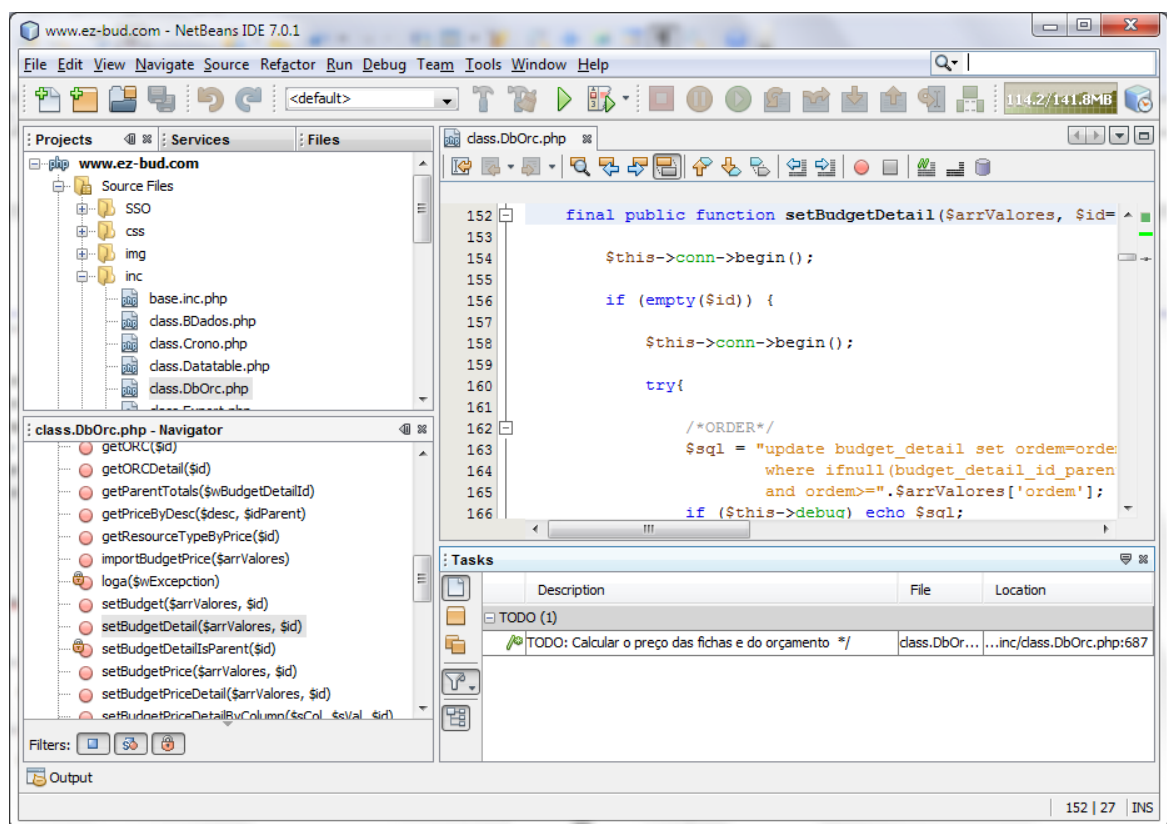


Figura 19: Ecrã da aplicação Netbeans

O sistema de licenciamento é duplo. É disponibilizado com licença 'Common Development and Distribution License (CDDL)' e GPL Versão 2 com exceção Classpath.

Disponível em <http://www.netbeans.org>

---

13 Integrated Development Environment

## 5.4 - PHPDOC

O PHPDOC é a norma corrente para auto-documentação da linguagem de programação PHP. Similar ao Javadoc mas escrita para PHP. No caso desta dissertação foi usada para a óbvia documentação e integrada no @Netbeans.

Disponível em <http://www.phpdoc.org/>

## 5.5 - HeidiSql

O HeidiSQL é um IDE para bases de dados MySQL. Permite explorar e todas as operações DDL e DML assim como operações de importação e exportação de dados para e de diversos formatos.

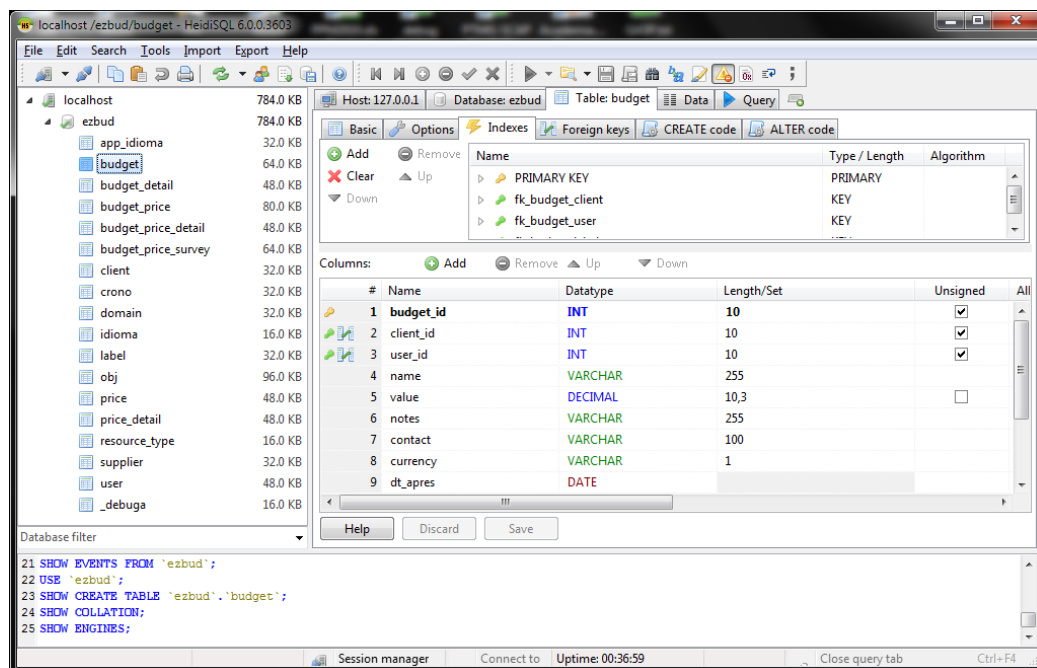


Figura 20: Ecrã da aplicação HeidiSQL

É uma aplicação de código aberto e o sistema de licenciamento é GPL Versão 2. Disponível em <http://www.heidisql.com>

## 5.6 - LibreOffice

O LibreOffice é um pacote de aplicações de produtividade destinado a utilizações profissionais ou pessoais. É compatível com as principais aplicações de produtividade no mercado e oferece todas as funções esperadas de um pacote profissional:

processador de textos, folhas de cálculo, apresentações, editor de desenhos, bases de dados e criação de fórmulas. Também permite a programação na linguagem simples BASIC.

O LibreOffice é a alternativa gratuita às aplicações tradicionais do ®Microsoft Office. É praticamente 100% compatível com os ficheiros criados no ®Word, ®Excel, ®PowerPoint, etc. Converte documentos criados no ®Microsoft Office para o LibreOffice, e vice-versa.

Inclui ainda muitas outras funções integradas, tais como a exportação direta para PDF, extensões, dicionários, impressão em série.

O LibreOffice está disponível para todos os principais sistemas operativos: ®Windows (Xp, Vista, 7), Linux (32 e 64 bits, com pacotes .deb e .rpm) e ®MacOS-X (processadores ®Intel e ®PowerPC).

O LibreOffice é disponibilizado sob os termos da licença pública LGPL, o que significa que todo o software pode ser alterado, personalizado e copiado.

Disponível em <http://pt.libreoffice.org/>

### **5.7 - GanttProject**

O GanttProject é uma aplicação multi-plataforma para gestão e calendarização de tarefas. É de distribuição gratuita e de código aberto e foi usado para a calendarização do trabalho assim como para os testes de exportação dos ficheiros em formato MPX.

### **5.8 - Mozilla Firefox**

Dispensa apresentações. É um excelente navegador para desenvolvimento uma vez que é extremamente completo dado as inúmeras extensões que auxiliam o programador, tanto na depuração como na análise da aplicação que está a desenvolver.

### **5.9 - Firebug**

Extensão do Firefox permite fazer a depuração do código de cliente, tanto o HTML, CSS e Javascript. Ver Figura 21.

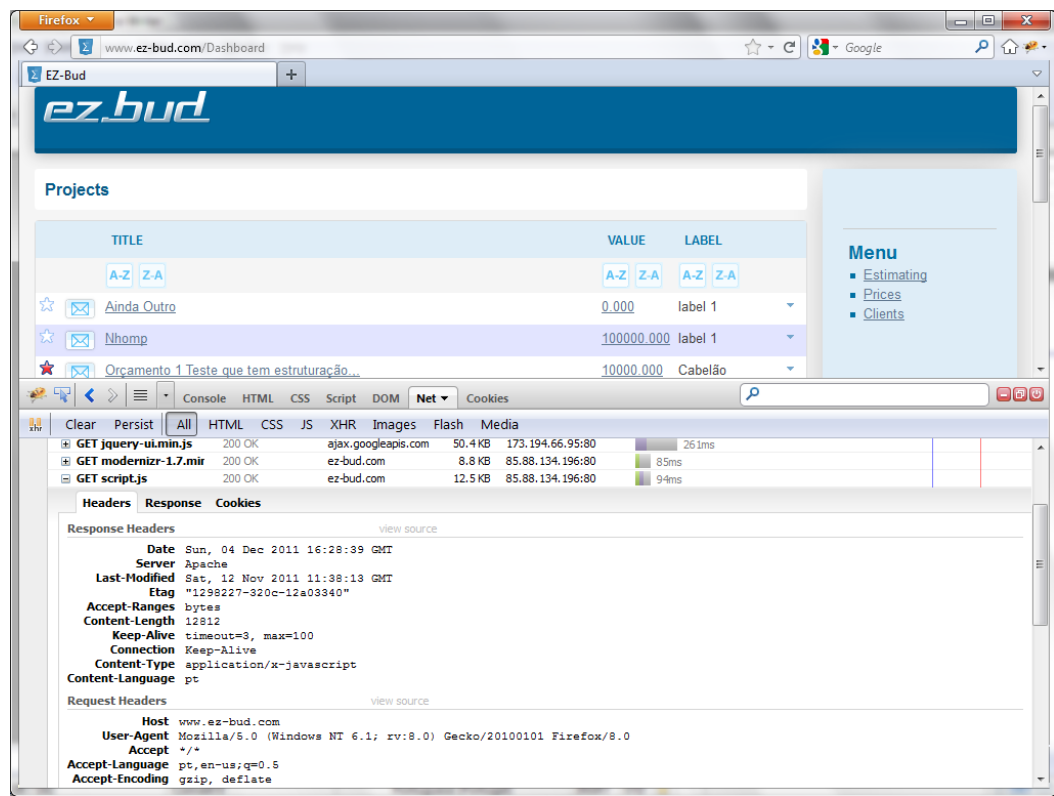


Figura 21: Firebug em depuração

## 6 - Testes

Os testes foram efetuados sem incidentes de grande monta, tendo mesmo ultrapassado as expectativas.

### 6.1 - Implementação

Relativamente a testes sobre o código desenvolvido, com auxílio de alguns ficheiros previamente desenvolvidos com as operações mais comuns, foram detetados problemas de pouca importância que foram de imediato resolvidos, não sendo dignos de menção.

### 6.2 - Carga

Foram efetuados testes de carga através da execução mais exigente para a Base de Dados, isto é o recálculo de um orçamento. Tendo em consideração o alojamento atual, foram executados 100 acessos em simultâneo vezes 10000 acessos e, se bem que um núcleo do processador estava ocupado a 100%, em momento algum a aplicação deixou de responder sem problemas. Foi utilizado o comando 'ab' do servidor Apache para o teste de carga:

- 'ab -n 10000 -c 100 http://www.ez-bud.com/stress.php'

Os resultados obtidos foram os seguintes:

#### 6.2.1 - Dados de acesso

Servidor: Apache

Nome do servidor: www.ez-bud.com

Porto de acesso: 80

Caminho do documento: /stress.php

Tamanho do documento: 3581 bytes

Nível de concorrência: 100

Tempo de execução dos testes: 116.157seg.

Pedidos completados: 10000

Pedidos falhados: 0

Erros de escrita: 0

Total de dados transferidos: 39760000 bytes

HTML transferido: 35810000 bytes



Média de pedidos por segundo: 86.09seg.

Tempo médio por pedido: 1161.569ms.

Rácio de transferência: 79.37Kb/seg.

### 6.2.2 - Tempos de ligação (ms)

	Mínimo	Média	Mediana	Máximo
Ligação	0	11	16	3011
Processando	94	1145	1079	4446
Esperando	78	899	889	4430
Total	94	1156	1092	4462

*Tabela 5: Testes - tempos de ligação*

### 6.2.3 - Percentagem dos pedidos servidos por período de tempo(ms)

50%	1092
66%	1127
75%	1170
80%	1186
90%	1264
95%	1388
98%	1685
99%	4040
100%	4462 (pedido mais longo)

## 7 - Conclusão

O desenvolvimento da aplicação Ez-Bud foi necessário uma vez que as soluções existentes não compreendiam por completo os requisitos elencados. Estes requisitos foram especificados tendo em consideração o paradigma vigente que é fortemente impulsionado pela adoção de dispositivos móveis e pela necessidade de ter acesso a informações em qualquer altura e lugar.

Por outro lado, a adoção de dispositivos móveis veio impulsionar a adoção do HTML5 e, no caso de novas aplicações, é imperativo que estas o comecem a adotar, mesmo tendo em consideração que ainda é uma norma em especificação e que ainda não esteja perfeitamente definida.

O desenvolvimento decorreu com eficácia e a utilização das API e bibliotecas selecionadas foi relativamente célere. No entanto, nem sempre é fácil conciliar as diversas bibliotecas o que leva a alguma criatividade e, nalguns casos, a implementação de métodos que permitam a interação entre as diversas bibliotecas e tecnologias. Relativamente à documentação das bibliotecas é, por vezes, escassa o que origina a leitura do código para melhor entendimento da ideia do programador da biblioteca, o que não deveria acontecer. Quanto às API Google, nenhum problema a relatar sendo extremamente fácil a interação, quer pela documentação completa quer pelos exemplo publicados para o efeito.

A decisão de utilizar uma plataforma de distribuição como a Google Apps é, a meu ver, uma boa solução para que haja mais visibilidade e uma rápida adoção, pois é possível disponibilizar a solução desenvolvida para um largo leque de possíveis clientes que são todos os subscritores do Google Apps.

A adoção de software livre ou de código aberto provou, mais uma vez, ser extremamente versátil não ficando atrás de nenhuma tecnologia dita como “proprietária”, com o benefício adicional que poderão haver implementações futuras que não estão dependentes de um Sistema Operativo específico.

O Ez-Bud ainda tem um longo caminho a percorrer uma vez que, de momento, ficará em testes, versão beta, durante alguns meses com o objetivo de se obter a melhor depuração possível, enquanto se desenvolvem mais algumas funcionalidades relativas ao ciclo de vida da própria obra, como o controlo de custos e a faturação, no entanto, não obstante o Ez-Bud ter enfoque na construção, poderá ser facilmente implementado noutras área que necessitem de elaborar orçamentos e controlo de custos.

## 8 - Bibliografia

- CCS – Clientes principais. <http://timelink.pt/empresa/utilizadores-principais/>. Acedido em 30 de Novembro de 2011.
- API Directory – ProgrammableWeb. <http://www.programmableweb.com/apis/directory>. Acedido em 30 de Novembro de 2011.
- Pressman, Roger S. (2001). Software engineering: a practitioner's approach. 5th ed. McGraw-Hill
- Epps, Sara Rotman, Forrester Research, (2010). The US Consumer PC Market in 2015, [http://www.forrester.com/rb/Research/updated\\_q3\\_2010\\_us\\_consumer\\_pc\\_market/q/id/57626/t/2?src=RSS\\_2&cm\\_mmc=Forrester\\_-\\_RSS\\_-\\_Document\\_-\\_24](http://www.forrester.com/rb/Research/updated_q3_2010_us_consumer_pc_market/q/id/57626/t/2?src=RSS_2&cm_mmc=Forrester_-_RSS_-_Document_-_24).
- Cohn, Mike (2005). Agile Estimating and Planning. Prentice Hall
- Goodman, Denny. (2001). JavaScript Bible, Gold Edition. Hungry Minds
- Cascading Style Sheets (CSS) Snapshot 2010. <http://www.w3.org/TR/CSS/>. Acedido em 30 de Novembro de 2011.
- Browser Support of CSS Properties & Selectors. <http://csscreator.com/properties>. Acedido em 30 de Novembro de 2011.
- HTML5. <http://www.w3.org/TR/html5/>. Acedido em 30 de Novembro de 2011.
- ECMA International (2011). ECMAScript Language Specification – Standard ECMA-262. <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>. Acedido em 30 de Novembro de 2011.
- PHP Documentation Group, (1997-2011). <http://www.php.net/manual/en/>. Acedido em 30 de Novembro de 2011.
- PHP-GTK: Documentation. <http://gtk.php.net/docs.php>. Acedido em 30 de Novembro de 2011.
- Codd, Edgar Frank (1985), "Does Your DBMS Run By the Rules", ComputerWorld.
- MySQL 5.1 Reference Manual. (2011). <http://dev.mysql.com/doc/refman/5.1/en/index.html>. Acedido em 30 de Novembro de 2011.
- Information Technology - Database Language SQL (Proposed revised text of DIS 9075). <http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt>. Acedido em 30 de Novembro de 2011.
- Information Management: A Proposal(1989), <http://www.w3.org/History/1989/proposal.html>. Acedido em 30 de Novembro de 2011.

- Apache HTTP Server Version 2.2 Documentation. (2011). <http://httpd.apache.org/docs/2.2/>.  
Acedido em 30 de Novembro de 2011.
- Datatables, Javascript API. (2007–2011). <http://datatables.net/api>. Acedido em 30 de  
Novembro de 2011.
- jQueryUI, Demos & Documentation. (2010). [http://jqueryui.com/docs/Getting\\_Started](http://jqueryui.com/docs/Getting_Started).  
Acedido em 30 de Novembro de 2011.
- jQuery JavaScript Library (English). (2010). <http://docs.jquery.com>. Acedido em 30 de  
Novembro de 2011.
- Google Apps APIs Overview. (2011). <http://code.google.com/googleapps/docs/>. Acedido em  
30 de Novembro de 2011.
- Zend Framework, Programmer's Reference Guide. (2006–2011).  
<http://framework.zend.com/manual/en/>. Acedido em 30 de Novembro de 2011.
- Description of the MPX Project File Exchange Format(2011).  
<http://support.microsoft.com/kb/270139/en-us>. Acedido em 30 de Novembro de 2011.
- Axure Software. <http://www.axure.com/download>. Acedido em 30 de Novembro de 2011.
- NetBeans IDE. <http://www.netbeans.org>. Acedido em 30 de Novembro de 2011.
- PHPDocumentor2. <http://www.phpdoc.org/>. Acedido em 30 de Novembro de 2011.
- HeidiSQL - MySQL made easy. <http://www.heidisql.com/>. Acedido em 30 de Novembro de  
2011.
- LibreOffice Portugal. <http://pt.libreoffice.org/>. Acedido em 30 de Novembro de 2011.