

[illegible]

foi possível frequentar um assim que entrei. Assim, grande parte das primeiras duas semanas foi passada a ler documentação, a desenvolver no génio (frequentemente à base de tentativa-e-erro), e a experimentar/manipular o código do próprio Genio, no que se espera que qualquer técnico da **ID! (ID!)** tenha um conhecimento fluente.

## 2.2 Práticas, Métodos, e Pipeline de Produção

Depois das tarefas que mencionei, foi necessário muito mais tempo do que imaginava para entrar no ritmo. O **processo** de produção empresarial é bastante mais complexo do que eu esperava, apesar de ter frequentado as cadeiras Engenharia de Software e Gestão de Projectos Informáticos. Habituei-me a utilizar um sistema interno de gestão de recursos humanos, e outro de gestão de projectos, ambos complementados por reuniões semanais de **SCRUM** e relatos frequentes ao coordenador da secção (e também **CTO! (CTO!)** da empresa) sobre os progressos obtidos. É utilizado o sistema **SVN** na empresa, para controlo de versões, e o Jenkins para integração contínua. As instalações são feitas directamente no cliente, dada a complexidade do sistema. Não me coube nenhuma instalação, visto não fazer parte das tarefas com que me comprometi, e não tenho os *skills* necessários para tal.

## 3 TRADUÇÃO C#/JAVA

Este foi o projecto para o qual fui contratado, e que me comprometi a concluir. No entanto, continua em desenvolvimento, pelo que os factos que aqui menciono serão possivelmente invalidados no futuro. Ainda assim, tentarei manter as conclusões sólidas e ser o mais concreto possível.

Para entender o conceito é necessário primeiro ter uma ideia, por geral que seja, de como funciona o motor de geração do Genio. Depois de desenvolver, ou antes descrever, visualmente, o programa a ser produzido no Genio, são escolhidas a plataforma- e linguagem-alvo. Cada alteração, ou detalhe (visual ou textual) que o utilizador criou é transformado num conjunto

de **definições** do Genio, que é armazenado numa base de dados local para subsequente utilização. Quando é desencadeada a geração, estas definições são aplicadas a um conjunto de *templates* (i.e. esqueletos de código-fonte), que, antes da geração não são utilizáveis, mas que, depois de aplicados ao domínio das definições do cliente se transformam em código fonte compilável, que compõe a solução final produzida.

Estes *templates* são compostos por código-fonte da plataforma em questão (e.g., C#, C++, ASP.NET MVC, etc.), encapsulados em instruções *NVelocity*. A *framework* *NVelocity* é um motor de *templates* de uso generalista. Ou seja, consiste num motor de geração e ficheiros (os *templates*) criados pelo utilizador, os quais são "realizados" em *run-time*. Assim se processa a geração de produtos no Genio.

### 3.1 Projecto Inicial - Expressões Regulares

Quando entrei na Quidgest foi-me entregue um prova de conceito moribunda, realizada anteriormente por um colega, que se revelou totalmente inadequada ao domínio do problema. Consistia em percorrer o código dos templates como texto, sem a mínima estruturação ou noção da sintaxe e semântica do código-fonte, e substituir determinadas expressões regulares por outras.

Depois de dois meses a insistir nesta aproximação, por sugestão da direcção, fiz finalmente um relatório de análise tecnológica a analisar a solução actual, os seus problemas, e as razões pelas quais o projecto nunca cumpriria o objectivo. Assim terminou esta primeira iteração do projecto.

*Terminou como?  
Para que serviu o relatório?*

### 3.2 Segunda Iteração - Tradução Manual

Sou bastante fluente em Java, já que sempre experimentei e pratiquei em casa, além do exigido pelas unidades curriculares do **IST! (IST!)**, porque sempre tive um grande interesse na linguagem e tecnologias envolventes. Assim, por sugestão minha, e com o apoio do meu coordenador, foi decidido que iria tentar fazer uma tradução manual dos *templates* do Genio. Assim comecei a **tarefa interminável**, sem o saber. A dimensão e complexidade do Genio

revelaram-se avassaladoras. Além disso, só comecei a utilizar a linguagem C# e a plataforma .NET quando entrei na empresa, e nesta altura tinha ainda pouca experiência. Durante dois meses insisti numa má aproximação, uma tradução quase linha-a-linha do código, em vez de me focar nos componentes e blocos lógicos, e tentar arranjar uma forma equivalente em Java, por diferente que fosse o código para tal. Depois destes dois meses fiz um relatório a admitir a derrota, e a explicar que não conseguiria traduzir a totalidade do Genio com este método. Precisava de tempo para estudar a estrutura deste, e a miríade de tecnologias utilizadas. Além disso, esta ideia derrotava o objectivo inicial, que era criar um tradutor **automático** que poupasse recursos na manutenção de outra plataforma (e consequentemente outras centenas de *templates*), o Java. Desta forma se encerrou a segunda tentativa de tradução, desta vez manual.

### 3.3 Progresso - Parser e Árvores Sintáticas

Apesar da "derrota" no projecto anterior, a direcção e coordenação acabou por aceitar a minha proposta: a utilização de um *parser* que gerasse *árvores sintáticas*, e a subsequente transformação destas árvores. Este tema foi abordado durante a totalidade da unidade curricular de Compiladores, do curso de Engenharia Informática, e, apesar de não ter tido uma nota brilhante estou bastante à-vontade com o tema, e tenho conhecimentos suficientes para investigar e desenvolver sobre o tema de uma forma autónoma.

Comecei então uma análise tecnológica, que durou uma semana, sobre as possíveis ferramentas e alternativas a utilizar, para maximizar a reutilização de código. Foi um sucesso. O projecto está atrasado, mas neste momento eu e o meu estagiário (que me foi "atribuído" há cerca de 3 meses, para ajudar no projecto) conseguimos, automaticamente, traduzir soluções inteiras de Visual C# automaticamente, e com os equivalentes tecnológicos funcionais de Java. Este *milestone* foi atingido esta semana, dia 27/05/2014. Note-se que perdi duas semanas de trabalho porque tive que dar formação ao estagiário, que nunca tinha utilizado C# e

não fazia ideia do eram parsers nem árvores sintáticas. Apesar de tudo, é uma pessoa extraordinária, dominou o tema rapidamente e é muito eficiente a trabalhar. Posso afirmar que acelerou substancialmente o ritmo de desenvolvimento do projecto.

## 4 TAREFAS, FERRAMENTAS E FORMAÇÕES

Apesar de estar há um ano na Quidgest, ainda hoje me vejo forçado a manter a par de tecnologias novas, e a dominar novas ferramentas, (e, às vezes, aprofundar as antigas). É difícil descrever este aspecto sem entrar numa interminável espiral técnica, mais semelhante a um curriculum vitae do que a um relatório de actividade. Assim, enumero as ferramentas que domino desde que entrei na empresa:

Linguagem C#, Visual Studio 2010/2012/2013, SQLServer, SMSQL, Java, Java EE, Ant, Maven, Eclipse, Resharper, Android SDK.

Relativamente às tecnologias que tive que aprender, seguem-se:

Expressões regulares em C# e Java, LALR parsers, Abstract Syntax Trees, NRefactory, NVelocity, .NET ADO, Windows Forms, WPF, WCF, ASP.NET MVC, JDBC, Java Security and Cryptography, Reflection.

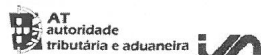
## 5 CONCLUSÃO

Esta foi uma actividade muito exigente em termos de tempo, já que continuo a estudar, e tenho outras actividades paralelas, mas apesar do "malabarismo" que tenho que fazer por vezes com os horários, tem corrido bem, e estou a conseguir desempenhar as tarefas que me foram entregues, se bem que o projecto esteja atrasado. No entanto, é um projecto de investigação, pelo que os atrasos perdem importância relativamente aos projectos de produção.

→ Neste tipo de documento (Técnico)  
a conclusão deve começar com  
um resumo do assunto tratado  
e depois mostrar o resultado!

## APÊNDICE

### COMPROVATIVOS DE EXECUÇÃO



Serviço de Finanças

SANTAREM - 2089

2014-01-24

IDENTIFICAÇÃO DO DOCUMENTO
54 37003829495
VALOR DE IDENTIFICAÇÃO FISCAL
215539745
VALOR A PAGAR (EUR)
1.782,50
PERÍODO
2014
TIPO DE PAGAMENTO
MENSAL <input type="checkbox"/> <input type="checkbox"/>
TRIMESTRAL <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
REGIME ESPECIAL DOS PEQUENOS RETALHISTAS
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
ACTO ISOLADO OUTROS OPERAÇÕES SUJEITAS
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

#### PAGAMENTOS DE IVA MODELO P2

Nome: JOSE MIGUEL GONÇALVES SIMÕES

Morada: R CAPITÃO ROMEU NEVES 2 2º ESQ.

SANTARÉM

Cód. Postal: 2005 - 157

SANTARÉM



Assinatura

VÁLIDO COMO RECIBO APÓS CERTIFICAÇÃO OU JUNTAMENTE COM O DOCUMENTO EMITIDO PELA ENTIDADE COBRADORA

54 1021 0003 6937003829495 0781

Isto não é comprovativo aduaneiro!

Se Melhete na empresa deve ter uma declaração dessa empresa!