

## Relatório Mini Projeto 1 Língua Natural - Grupo 9

José Guilherme Freitas Gomes Nº 89378

Gonçalo Maria Rodrigues Bustorff Silva Nº 82050

Iniciámos o projeto analisando os exemplos já fornecidos, visto nunca termos trabalhado com o *OpenFST*. Após compreensão do funcionamento do software decidimos, para motivos de facilitação na escrita dos transdutores, começar por desenhar todos os transdutores base. Após desenharmos e implementarmos manualmente os transdutores (*.fst*) *mmm2mm*, *misto2numerico*, *en2pt*, *pt2en*, *dia*, *mês* e *ano*, utilizámos as funções da ferramenta fornecida para implementarmos os restantes *numerico2texto*, *misto2texto* e *data2texto*. Testámos como requisitado (data de nascimento dos integrantes do grupo) e a solução apresentou os resultados esperados. Mas, após análise da solução e requerimentos/regras do enunciado, a nossa solução apresentava-se como demasiado complexa, pouco flexível e propensa a erros. Vários dos nossos transdutores tinham a possibilidade de se tornar mais simples caso fragmentados e gerados pelo *OpenFST*. Apenas traduziam datas e, por necessitarem de símbolos que não os requisitados, incorriam em erro. Tudo isto principalmente por assumirmos que o estado terminal de um transdutor termina a leitura de *input*, o que nos limitava imenso em termos de soluções alternativas às inicialmente desenhadas.

Após uma sessão de esclarecimento de dúvidas pela parte do profº Nuno Mamede, na qual foram mostradas as nossas verdadeiras possibilidades, decidimos repensar e reformular todo o projeto, desta vez tendo por objetivo implementar transdutores o mais simples possível (com o menor número de estados), que cumprissem com os requisitos do projeto.

Depois de reestruturado, o total de transdutores feitos manualmente aumentou para 8, mas a complexidade de cada um reduziu drasticamente. Apenas foram implementados o *dia*, *ano*, *mês*, *barra2barra*, *barra2de*, *en2ptaux*, *mmm2mm* e *num2num*. Os transdutores ***barra2barra*** e ***barra2de*** têm apenas 1 estado cada, pelo que a sua função é apenas ler e substituir “/” por “/” e “de”. O transdutor ***num2num*** lê e devolve o mesmo número pelo que também é constituído por apenas 1 estado. O ***mmm2mm*** lê o mês por extenso e devolve o numérico de dois dígitos correspondente, pelo que é constituído por um estado por cada mês mais o estado final. A tradução dos meses é feita no ***en2ptaux***, que lê o mês por extenso e retorna-o no mesmo estado, sendo que existe apenas o inicial. Apesar de o objetivo principal seja converter datas para extenso, o transdutor ***dia*** permite converter qualquer número no intervalo [1..99] para extenso, cumprindo-o sem a necessidade de adicionar estados ao necessário para os dias do mês ([1..31]). O ***mês*** necessita apenas de 2 estados para converter o(s) dígito(s) para extenso. Finalmente o ***ano*** converte os 4 dígitos pertencentes ao intervalo [2001..2099] e converte-os para extenso. Este último apresenta naturalmente uma maior complexidade, visto o maior número de dígitos e adições ao *output* (“e” entre conversões, nalgumas vezes necessária). Tendo estes transdutores descritos como componentes base, o ***misto2numerico*** foi gerado concatenando o *barra2barra*, o *mmm2mm* e o *num2num*, o ***pt2enaux*** invertendo o *en2ptaux*, o ***en2pt*** concatenando o *barra2barra*, o *num2num* e o *en2ptaux*, o ***pt2en*** com o *pt2enaux* similarmente, o ***numerico2texto*** concatenando o *dia*, *barra2de*, *mês* e *ano*, o ***misto2texto*** compondo o *misto2numero* e o *numerico2texto* e finalmente o ***data2texto*** fazendo a união do *misto2texto* com o *numerico2texto*. Todos os testes foram bem-sucedidos pelo que consideramos esta uma excelente solução cuja qual estamos muito satisfeitos que, desta vez, executa tudo o requisitado sem recorrer a transdutores escritos desnecessariamente, sendo que todos os implementados são simples e flexíveis.