

projetos universitários, com regras pré definidas, que apresentam um caminho linear para a sua realização. Agora deparo-me com um amplo conjunto de ferramentas que facilitam a programação, em que há espaço para analisar opções e desenvolver um pensamento crítico relativo aos prós e contras de cada uma.

Foi decidido que íamos usar javascript para o servidor utilizando a plataforma *node* (<http://nodejs.org>), que serve de base para construir aplicações distribuídas. Esta plataforma permite-nos ter vários clientes a aceder ao servidor, sem atrasos no tempo de resposta, e um fácil escalonamento, caso seja necessário permitir um maior número de utilizadores concorrentes. Como o cliente também é javascript, não há necessidade de programar em linguagens diferentes. Estas decisões vão nos permitir diminuir o tempo de desenvolvimento. A comunicação entre os dois dá-se com pedidos HTTP, que é um tipo de mensagem usada na internet. As mensagens HTTP usam verbos para declarar o tipo de pedido do cliente: GET, POST, UPDATE, DELETE e PUT, o servidor fica à espera destes pedidos em endereços diferentes, processa o pedido e envia uma resposta.

3.2 Base de Dados

Na base de dados vão ser guardados os objetos que têm de ser persistentes. Optou-se por usar *MongoDB*, porque assim o servidor tem acesso a funções de alto nível para interacção com a base de dados, mas os conhecimentos teóricos, que eu aprendi no curso, não são todos os mesmos. Com *MongoDB*, em vez de agrupar os objetos em tabelas, como em *MySQL*, cada objeto é um documento independente dos outros e o código do servidor tem uma sintaxe mais simples e clara para interagir com a base de dados.

A base de dados está dividida em utilizadores, ficheiros e comentários. Nos utilizadores guarda-se informação pessoal para poderem aceder e interagir com a aplicação, nos ficheiros guardam-se *links*, que permitem ao servidor redireccionar o utilizador para a página com o ficheiro correspondente e nos comentários guarda-se, como o próprio nome indica, os comentários de todos os utilizadores.

Houve ainda uma preocupação na forma como são guardadas as palavras passe, que, supondo que um atacante consiga acesso à base de dados e as roube, têm de ser cifradas, para que não seja possível acesso indevido às contas dos utilizadores.

Nod. no serve 4 esta informação com SOFT. Skill

4 LÓGICA DE NEGÓCIO

Com a base de dados implementada é necessário definir a forma como os utilizadores vão poder interagir com ela, a isto dá-se o nome de lógica de negócio. A ideia principal é que o utilizador pode aceder a um endereço específico com certos atributos e alterar o estado da base de dados. Como exemplo, o utilizador pode querer criar um conta, do lado do servidor são recebidos os dados e são verificadas as regras impostas, se os dados forem aceites, a base de dados é acedida e os dados ficam guardados persistentemente. Estas regras impedem que seja criado um utilizador com um nome já existente, ou que a palavra passe não tenha o tamanho certo. As regras têm de ser fortes o suficiente para impedir qualquer tipo de uso abusivo por parte de utilizadores mal intencionados.

Nod. no serve 5 esta informação com SOFT Skill

5 TESTES

Como falei no tópico anterior, o sistema tem de ser capaz de detetar a informação errada que recebe dos utilizadores e verificar o funcionamento correto do código. Mas os casos possíveis são tantos, que testar o sistema manualmente demoraria demasiado tempo, e qualquer alteração ao código do sistema obrigaria a testar tudo de novo. É necessário criar um conjunto de testes que podem ser corridos automaticamente, que enviam mensagens ao servidor e verificam as suas mudanças de estado. Assim podemos ter um maior grau de confiança de que o sistema funciona de maneira correta.

6 CONCLUSÃO

A realização desta atividade consciencializou-me, de como é o trabalho de um programador, e das várias dificuldades que eu vou sentir quando acabar o curso. O processo de desenvolvimento de uma aplicação é longo

autor?

e imprevisível, trabalhei cerca de 50 horas, aprendendo novos conhecimentos sobre bases de dados e aplicações distribuídas, estudando as ferramentas a usar e aplicando estes novos fundamentos. Sem quaisquer conhecimentos da linguagem javascript consegui aprender por mim próprio, aplicando os mesmos conceitos de outras linguagens de programação que já conhecia. Aprendi a usar as várias ferramentas, ainda que de um modo superficial, para poder usá-las no projeto. Depois do trabalho feito, adiantei bastante a minha parte, sentindo já uma certa facilidade a programar deste modo e com uma maior preparação para aprender novas linguagens e ferramentas no futuro. O projeto ainda se encontra longe de terminado, mas vou conseguir ser mais produtivo depois de tudo o que já fiz. Penso que ganhei uma maior espírito crítico relativamente às opções que tenho como programador e tenho uma melhor ideia da especialização que quero seguir.

Tudo isto são
⇒ HARD
SKILLS!

Este documento não o presente
nenhum. APRENDIZAGEM a nível
pessoal, em termos comportamentais, de relacionamento
de competências sociais, negociais, de organização, etc!