

Estágio de Programador de Software - Location Intelligence



Licenciatura em Eng. Informática
Estágio-Projeto 2020-2021



Vasco Miguel Batinas Barnabé

Orientador na empresa: Mário Sobral

Orientador no departamento: Professor Francisco Coelho

Trabalho desenvolvido na empresa Focus BC no âmbito da disciplina de Estágio-Projeto da Licenciatura em Eng. Informática.

Évora, 25 de junho de 2021

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	1
1.3	Contribuições	2
1.4	Estrutura do documento	2
2	Ambiente empresarial	3
3	Ambiente de desenvolvimento	4
3.1	Ambiente técnico	4
3.1.1	Software	4
3.1.2	Ambiente aplicativo	4
3.2	Metodologia de trabalho	5
4	Trabalho desenvolvido	6
4.1	Descrição detalhada	6
5	Avaliação crítica	15

1 Introdução

A formação em contexto de trabalho proporciona aos alunos/formandos uma experiência prática em contexto real, determinante para a aplicação de conhecimentos e desenvolvimento de competências técnicas, relacionais e organizacionais relevantes para o perfil profissional do curso frequentado pelo aluno, sendo esta formação estruturada e orientada por um tutor.

Esta vertente do ensino permite ao aluno contactar com tecnologias e técnicas específicas de uma profissão, sendo uma oportunidade de aplicação dos conhecimentos adquiridos a atividades concretas da mesma, potenciando assim um impacto positivo na aprendizagem e no desenvolvimento das competências profissionais do formando.

Promove hábitos de trabalho, espírito empreendedor, sentido de responsabilidade profissional e o despertar para as relações humanas em contexto laboral. Permite sem dúvida vivências sobre a realidade do funcionamento das organizações, tendo como objetivo facilitar posteriormente a integração no mercado de trabalho.

Em paralelo, para a entidade ou instituição que acolhe os formandos também existem benefícios pois em alguns casos esta formação em contexto de trabalho pode facilitar futuros processos de recrutamento de pessoal, exigindo por parte dos formadores atualizações constantes de modo a poderem acompanhar os formandos.

Finalizada essa formação, é igualmente importante a realização de um relatório sobre a mesma de modo a que o formando reflita sobre o seu processo de aprendizagem, sobre as novas aquisições de conhecimentos e de competências e sobre a experiência que vivenciou; sendo este também um documento que irá espelhar o seu percurso no estágio junto do orientador da instituição de ensino na qual se encontra inserido.

1.1 Enquadramento

O presente relatório enquadra-se na formação em contexto de trabalho inserida no plano curricular do curso de "**Engenharia Informática**", ministrado na "Universidade de Évora".

A referida formação foi realizada na empresa Focus BC, sediada em Lisboa, com início a 8 de Março e término a 11 de Junho do corrente ano, com a duração de 312 horas. É importante referir que devido ao contexto pandémico, este estágio foi realizado de forma integral à distância, via online.

1.2 Objetivos

De modo a se obter uma noção mais precisa do estágio realizado, nesta seção serão apresentados dois tipos diferentes de objetivos: os objetivos desta formação em contexto de trabalho e posteriormente os objetivos do trabalho realizado durante esta formação.

A realização da formação em contexto de trabalho a que este relatório se refere, teve os seguintes objetivos:

- Desenvolver soluções simplificadas baseadas em Google Cloud, em ambiente web e com recurso às mais recentes tecnologias.
- Adquirir conhecimentos na plataforma Google no âmbito de Cloud e Maps, para aplicação nas soluções e tarefas alocadas.
- Aplicar conhecimentos e técnicas de programação, com acesso à plataforma Mapify e respetivas funcionalidades de Location Intelligence, para a criação de soluções funcionais que demonstrem a aquisição de conhecimento do estagiário nas técnicas, ferramentas e processos de desenvolvimento de software.

No que diz respeito ao objetivo do trabalho desenvolvido durante a formação em contexto de trabalho, este foi:

- Criar uma "quickstart solution for Internet of Things" (um simulador de sensores capaz de reunir e transmitir dados), baseada na plataforma Mapify.

1.3 Contribuições

Ao longo destas 14 semanas de estágio na empresa em questão, foi sendo construído um simulador de sensores. O orientador deste estágio referiu no final que iria passar a utilizar esta ferramenta de testagem em diversas funcionalidades de aplicações da empresa.

Foi ainda criada uma "Client application for Sensor Data Visualization" que nos permite visualizar em tempo real num mapa, a localização dos sensores que foram ativados.

1.4 Estrutura do documento

Durante o desenvolvimento deste relatório será descrita a forma como foi criado o simulador, quais as suas funcionalidades, as tecnologias utilizadas, as dificuldades que surgiram durante a realização do mesmo, as soluções encontradas e os conhecimentos adquiridos ao longo desta experiência.

Posteriormente será descrita a aplicação de visualização de sensores construída, qual o objetivo da sua utilização, e os resultados obtidos.

2 Ambiente empresarial

Tal como já foi mencionado, a empresa Focus BC tem sede em Lisboa, criada em Fevereiro de 2012 e especializada na área da inteligência de localização. É composta por uma equipa dinâmica, especializada em diversas áreas, desenvolvendo inúmeros projetos para os mais diversos clientes, nacionais e internacionais. É importante referir a sua parceria com a Google a nível de Google Cloud e da plataforma Maps.

A empresa em questão, funciona com equipas afetas a diferentes projetos. No nosso primeiro contacto, foi-nos proposto ingressarmos em duas equipas distintas, destacando dois formandos para cada equipa, uma vez que desta instituição de ensino fomos quatro alunos a efetuar a formação. Em parceria com outro colega de curso, fui inserido na equipa responsável pelo projeto **Mapify** (plataforma de inteligência de localização para utilização empresarial), e em conjunto desenvolvemos ideias de modo a dar resposta às solicitações do orientador da empresa, e aos objetivos deste estágio. O profissional destacado para nosso orientador foi um dos elementos que realizou a apresentação da empresa e do estágio proposto aquando a divulgação da mesma junto dos alunos desta universidade. O seu nome, Mário Sobral. Foi este o elemento que nos acompanhou mais de perto durante estas 14 semanas; no entanto, também tínhamos acesso a outros elementos da equipa, que sempre se mostraram disponíveis para as nossas questões.

Um aspeto que me agradou bastante durante a realização deste estágio foi poder observar de perto o espírito de equipa e o ambiente informal presente entre todos os elementos desta empresa, e a forma calorosa como fomos acolhidos e integrados. Senti-me a fazer efetivamente parte deste mundo, desta equipa, e desta empresa.

3 Ambiente de desenvolvimento

A primeira semana de estágio, por sugestão do nosso orientador, Mário Sobral, decorreu com uma fase de onboarding com acesso a material de estudo da Google no âmbito de Cloud e Maps, para aplicação nas soluções e tarefas alocadas com o objetivo de realizar uma aprendizagem introdutória de Google Maps e Google Cloud e de conceitos de Geo Location.

3.1 Ambiente técnico

3.1.1 Software

Na área da informática, a utilização de ferramentas de programação é imprescindível e da sua conjugação surgem aplicações que procuram dar resposta às necessidades do cliente. De acordo com os objetivos de cada projeto, assim são selecionadas essas ferramentas.

Para a criação do simulador de sensores e da web application, foram utilizadas as seguintes ferramentas de programação:

- **NodeJS** - ambiente de execução utilizado para o desenvolvimento do simulador de sensores, podendo este simulador ser executado como uma aplicação "standalone" numa máquina, tendo sido esta a primeira vez que tive contacto com a ferramenta;
- **Typescript** - linguagem de programação escolhida para o desenvolvimento do simulador de sensores. Embora esta tenha sido a primeira vez que programei com a linguagem, senti que me adaptei facilmente às suas características, uma vez que esta tem muitas parecenças com Javascript, linguagem com a qual já havia programado em contexto académico;
- **Javascript** - linguagem adotada para a criação da Sensor Data Visualization Application e de um servidor a esta associado; não senti grandes dificuldades a programar com a linguagem, pois, como já foi referido, já havia programado com ela em contexto académico e nesta altura do estágio, já vinha com alguma prática na utilização de Typescript para a criação do simulador, tendo sido por isso relativamente fácil programar com esta linguagem;
- **HTML/CSS**(HyperText Markup Language / Cascading Style Sheets) - esta linguagem de programação e este mecanismo foram utilizados na criação e definição de estilo da página web criada para visualização de sensores; à semelhança da linguagem de programação Javascript, estas ferramentas também já haviam sido por mim utilizadas anteriormente;
- **Google Maps Javascript API** - serviço que nos permitiu adicionar e personalizar uma mapa do Google na Web App desenvolvida, usando Javascript;
- **Google Cloud** - ambiente de computação em nuvem da Google, utilizado para a criação de uma VM (Virtual Machine) onde o servidor web criado para a Sensor Data Visualization Application foi executado;
- **Jira** - ferramenta que permite a monitorização de tarefas e o acompanhamento dos projetos, gerenciando todas as atividades em simultâneo; foi a primeira vez que utilizei este tipo de ferramenta de modo a organizar o desenvolvimento do projeto proposto e com a qual me identifiquei bastante;
- **Bitbucket** - sistema de controle de versões distribuído baseado em Git utilizado no desenvolvimento de todas as aplicações desenvolvidas neste estágio.

Além de todo este software já mencionado anteriormente, é importante fazer referência a algumas bibliotecas ou frameworks de **NodeJS** utilizadas neste estágio, que irão ser referidas de forma mais detalhada num capítulo mais à frente, neste relatório, nomeadamente: **MQTT.js** (biblioteca do protocolo MQTT escrita em Javascript para NodeJS) e **Express.js** (framework para NodeJS).

3.1.2 Ambiente aplicacional

No que diz respeito ao contexto aplicacional onde o trabalho realizado se enquadra, considero que se trata de uma aplicação "standalone", isto é, estamos perante uma aplicação autossuficiente.

Este simulador, de forma a ser utilitário interage com a plataforma Mapify e também com a Web App desenvolvida para visualização gráfica dos sensores, completando-as. A sua interação é descrita num capítulo posterior.

3.2 Metodologia de trabalho

Diariamente eram realizadas standups em que todos os elementos da equipa na qual fui inserido, se reuniam debatendo os progressos do seu trabalho no dia anterior, planificando também o trabalho a ser realizado nesse dia. Enquanto formandos, apenas participávamos nas standups de segundas e terças feiras (não participávamos nas restantes por uma questão de conciliação de horários com as aulas na universidade) e também nós apresentávamos os nossos progressos.

Em simultâneo, eram realizados sprints quinzenais da empresa, designados por “Bi-Weekly”, contando com a presença de todos os profissionais, e nós, formandos incluídos. Nestas reuniões, os líderes das equipas e os responsáveis por desenvolvimento de projetos, apresentavam de uma forma breve e sucinta, se os objetivos definidos para aquele sprint tinham sido ou não atingidos, os progressos conseguidos, e os obstáculos enfrentados. No final, eram definidos novos objetivos a atingir até ao próximo sprint.

Em relação ao nosso percurso durante estas 14 semanas, foi adotado um modelo de planeamento e de desenvolvimento de software em ambiente **Agile**. Este modelo caracteriza-se por dividir cada projeto em pequenas partes, a serem realizadas em determinados períodos de tempo, curtos, de modo a simplificar a compreensão de prioridades e a introdução de mudanças no desenvolvimento do projeto, caso seja necessário. Em suma, trata-se de uma abordagem interativa, destinada a desenvolver projetos organizados segundo os objetivos e necessidades dos clientes. O software utilizado foi o **Jira**, ferramenta que permite a monitorização de tarefas e o acompanhamento dos projetos, gerenciando todas as atividades em simultâneo.

No **Jira** foi criado um Board dividido em seções, onde consoante o estado de desenvolvimento de cada tarefa, assim esta era alocada ao separador que designava o seu estado (To Do/ Blocked / In Progress / Review / Done). Ao longo do desenvolvimento do projeto, de acordo com a fase em que este se encontrava, foram criados backlogs (lista de tarefas ou pendências necessárias como suporte do plano estratégico, tratando-se de um repositório de requisitos a serem desenvolvidos e entregues). Organização de trabalho que considere muito positiva, pois deste modo, todos os elementos envolvidos (neste caso, éramos dois formandos a desenvolver o mesmo projeto), tinham conhecimento em tempo real das tarefas a desenvolver, em desenvolvimento ou já realizadas. Outro aspeto muito importante foi o facto de existir um separador (Blocked) onde poderíamos colocar as tarefas estagnadas, que apresentavam bugs por resolver, permitindo aos restantes elementos da equipa contribuírem para a sua resolução. O orientador era o responsável pela avaliação da realização de cada tarefa, atribuindo-lhe a designação de “Done” sempre que esta estivesse terminada e fosse por si aceite. Este modo interativo de realização de trabalho permitiu-nos, para além de desenvolver o espírito de equipa e de entreaajuda, realizar este projeto em comum, em que ambos poderíamos desenvolver qualquer uma das tarefas, complementando-nos e aperfeiçoando o trabalho desenvolvido pelo colega. Deste modo, é impossível referir quais as tarefas realizadas apenas por cada um dos formandos. Foram debatidas ideias, realizadas tarefas e obtidos resultados complementares.

Como sistema de controle de versões para o desenvolvimento do nosso software, foi utilizado o **Bitbucket** (ferramenta de hospedagem e colaboração de código baseada em **Git**, criada para equipas profissionais de engenharia de software e gestão de projetos). É este o software **Git** utilizado pela empresa Focus BC.

4 Trabalho desenvolvido

Tal como já foi referido anteriormente, a primeira semana do estágio foi essencialmente composta por formação teórica sobre Google Cloud e a Plataforma Maps, de modo a obter conhecimentos base relacionados com o projeto que iria desenvolver. Tendo este estágio de programador de software como um dos objetivos a aplicação de conhecimentos e técnicas de programação, com acesso à plataforma Mapify e respetivas funcionalidades de Location Intelligence, verificou-se ser de extrema importância este tipo de aquisição de conhecimentos na fase inicial do estágio.

Na fase seguinte, e já integrado na equipa afeta ao projeto Mapify, sob a orientação do tutor Mário Sobral, demos início ao desenvolvimento do principal objetivo deste estágio, a criação de uma quickstart solution for IoT(Internet of Things), um simulador de sensores MQTT (Message Queuing Telemetry Transport), utilizando Typescript como linguagem de programação.

Nesta fase do relatório, julgo ser importante apresentar uma explicação sucinta acerca do funcionamento do simulador, de modo a que, posteriormente, a sua explicação detalhada seja de mais fácil compreensão. Este funcionamento consiste na criação de mensagens de X em X segundos (configuração dependente do utilizador) e que as mesmas sejam enviadas automaticamente para o Mapify (aplicação já referida anteriormente, criada pela Focus BC), onde os dados poderão ser apresentados num plano gráfico. Cada simulador está afeto a um sensor, tendo que gerar mensagens para simular esse mesmo sensor. Cada mensagem contém a localização do sensor e outros atributos, definidos pelo utilizador no template da mensagem, previamente definida num ficheiro de configuração.

Após a criação do simulador, foi agendada para o dia 27 de Abril do corrente ano, uma demo do simulador criado, para toda a empresa. Nesta reunião surgiram alguns comentários construtivos e algumas sugestões de melhoria. Atendendo às funcionalidades do nosso simulador, surgiram ideias entre os profissionais da empresa acerca de algumas alterações a efetuar à plataforma Mapify, após terem visto como esta se comportava perante o nosso simulador.

As semanas seguintes foram dedicadas a implementar melhorias, tendo sempre em conta as sugestões propostas na demo realizada.

Uma vez atingido o objetivo inicialmente proposto, e porque ainda restavam cerca de 3 semanas para o término do estágio, em conjunto com o tutor de estágio foi decidido que poderíamos avançar para a criação de uma Client Application for Sensor Data Visualization.

A finalidade desta aplicação assenta na visualização num mapa dos sensores ativos em movimento, ou seja, fazer com que a informação gerada no simulador de sensores criado, fosse apresentada para o cliente num mapa em tempo real. Também este objetivo foi atingido.

No final deste estágio, no dia 8 de Maio, foi realizada uma apresentação final perante todos os colaboradores da empresa e restantes colegas de estágio, encerrando assim o meu percurso enquanto aluno a realizar um estágio inserido no meu currículo académico.

4.1 Descrição detalhada

Uma vez que para o arranque deste simulador é necessário recorrer à linha de comandos, as primeiras tarefas realizadas basearam-se na leitura e certificação dos parâmetros de input inseridos pelo utilizador do simulador.

No entanto, é importante referir que este simulador foi pensado e implementado de forma a detetar qualquer falha por parte do utilizador no que concerne aos parâmetros inseridos por este e dos quais o bom funcionamento do simulador depende.

Deste modo, os parâmetros de input necessários para o arranque e funcionamento do simulador e por esta mesma sequência são:

- ID do sensor;
- ficheiro de configuração;
- ficheiro de configuração;
- intervalo de tempo entre o envio de cada mensagem;

-
- distância máxima entre localizações do sensor entre duas mensagens consecutivas.

Explicando um pouco em pormenor os parâmetros de input:

- **id do sensor:** o id do sensor é obtido como um parâmetro de input especificado pelo utilizador quando este arranca o simulador, para que o sensor possa ser identificado e distinguido de outros que estejam ativos.
- **ficheiro de configuração:** Tal como já foi referido anteriormente, as definições do simulador são configuradas através de parâmetros, introduzidos na linha de comandos pelo utilizador, aquando o arranque do mesmo.

No entanto, o conteúdo deste ficheiro de configuração(ficheiro em formato **JSON** - JavaScript Object Notation) é composto por um template da mensagem que o simulador irá enviar para o Mapify e por dados que identifiquem o **Data Feed** (mecanismo para os usuários receberem dados atualizados de fontes de dados, usualmente utilizado por aplicativos que funcionem em tempo real em configurações ponto a ponto) do Mapify para onde serão enviadas as mensagens.

A localização de um sensor será sempre um Ponto, representado por coordenadas geográficas (latitude e longitude).

É calculada a distância entre o ponto da mensagem anterior e o novo ponto para a nova mensagem, uma vez que essa é também uma propriedade (um atributo) do sensor, e que necessita de ser substituído na mensagem.

Outros atributos do sensor que o utilizador defina no ficheiro de configuração podem ser do tipo **string**, **numeric** ou **boolean**.

Estes atributos podem, ainda, ser de dois tipos: **estáticos** ou **dinâmicos**.

Estáticos: o valor deste atributo do sensor será sempre o mesmo em cada mensagem gerada, sendo composto por um valor do tipo definido para aquele atributo (string, numeric, boolean). exemplo: “driver”: “Rui”

Dinâmicos: o valor do atributo pode variar de mensagem para mensagem, sendo obtido de forma random entre as opções fornecidas pelo utilizador.

- **Boolean:** O valor deste atributo varia de forma random entre verdadeiro e falso;

Exemplo: “isLightOn”: true/false

- **String:** O valor deste atributo é escolhido de forma random entre uma série de opções que o utilizador fornece, num array de strings;

Exemplo: “color”: [“Red”, “Yellow”, “Blue”, “Green”] (em cada mensagem, o simulador escolherá uma destas quatro possibilidades para preencher o valor do atributo em questão, “color” neste caso)

- **Numeric:** o utilizador fornece os valores limite superior e inferior, e a cada mensagem gerada o programa atribui ao atributo um valor random que se enquadre entre os limites fornecidos. Exemplo: “temperature”: “[100,100]”;

ou

O utilizador fornece um conjunto de valores e o programa atribui ao atributo um dos valores existentes.

Exemplo: “temperature”: “[10,20,30,50,80]”,

O template da mensagem depende das informações que o utilizador pretenda que estejam contidas nesta. No entanto, o fim para o qual este simulador foi desenhado e implementado exige que nesta template esteja presente um objeto “location”, com um atributo “coordinates” onde está definida a Bounding Box (limites máximos de latitude e longitude que definem uma bounding box para o sensor) e a cada mensagem enviada o conteúdo deste atributo serão as coordenadas do sensor (latitude e longitude); um atributo “type”, sempre definido como “Point” e um atributo “distance”, que inicialmente no ficheiro de configuração se encontra vazio mas a cada mensagem irá conter a distância entre a última localização conhecida daquele sensor e a nova localização. A posição do sensor será alterada a cada mensagem enviada, estando essa localização sempre dentro da bounding box definida.

Exemplo de um ficheiro de configuração:

```
{
  "sensorMessage":
  {
    "sensor_id": "$$i $$",
    "vehicleType": "$$$ [Car,Truck,Bus,Motorcycle,Bicycle] $$",
    "driver": "$$$ [Mario,Fabio,Sandro,Rui,Vasco] $$",
    "speed": "$$n [0,100] $$",
    "temperature": "$$n [-10,100] $$",
    "maxTemperature": "$$n [-10,100] $$",
    "voltage": "$$n [0,20] $$",
    "location":
    {
      "type": "Point",
      "coordinates": [ "$$c [-9.369619, -9.124143, 38.712672, 38.795403] $" ],
      "distance": "$$d $$"
    }
  },
  "mqttBroker":
  {
    "username": "a883a1da-d7bd-4403-8ea1-95f25f9864a4",
    "password": "ae33f128ce27f4efb811425fa2e4b97bbf0ce7b2dde51d4263e79a3f5b51f957",
    "topic": "a883a1da-d7bd-4403-8ea1-95f25f9864a4/Car-Sensors",
    "url": "mqtt://broker-qa.mapify.ai",
    "port": "1883"
  }
}
```

Para se definir um valor estático, basta colocá-lo como valor do atributo, mas se o utilizador pretender que o valor de determinado atributo seja dinâmico, isto é, varie de mensagem para mensagem, necessita de seguir a configuração de tipos de valores que o simulador fornece, da seguinte forma: “\$(código) valor \$”.

Lista de Códigos disponíveis

- **n** (Array com dois valores, um máximo e outro mínimo) - para quando o utilizador pretende que o valor deste atributo seja um número entre o valor máximo e mínimo fornecidos - “speed”: “\$\$n [0,100] \$\$”;
- **s** (Array de Strings) - para quando o utilizador pretende que o valor seja uma palavra dos existentes no array - “driver”: “\$\$\$ [Mario,Fabio,Sandro,Rui,Vasco] \$\$”;
- **c** (Coordenadas) - o utilizador fornece a bounding box (valores mínimos e máximos de longitude e latitude) que delimitará o campo por onde o sensor pode movimentar-se - “coordinates”: [“\$\$c [-9.369619, -9.124143, 38.712672, 38.795403] \$”];
- **b** (Boolean) - o utilizador pretende que o valor deste campo seja verdadeiro ou falso, não sendo necessário inserir qualquer valor - “inMovement”: “\$\$b \$\$”;

-
- **m** (Array de números) - o utilizador pretende que o valor seja um número escolhido ao acaso de entre os existentes no array - “Goals”: “\$\$m [10,20,30] \$\$”;
 - **d** (Distância) - o valor deste campo será a distância entre a nova localização do sensor e a anterior, não sendo necessário inserir qualquer valor (o valor deste campo na primeira mensagem gerada será sempre 0) - ”distance”: ”\$\$d \$\$”;
 - **i** (ID do sensor) - este código foi criado uma vez que o ID do sensor é fornecido pelo utilizador como parâmetro de input e faz parte da mensagem a ser enviada. Deste modo, apesar do valor (id do sensor) ser sempre o mesmo, este campo é dinâmico pois a cada mensagem gerada o seu valor é sempre substituído, embora por um valor igual - ”sensor_id”: ”\$\$i \$\$”.
- **Nº de mensagens que serão enviadas** - este número é obtido como um parâmetro de input especificado pelo utilizador quando este arranca o simulador;
 - **Intervalo de tempo entre o envio de cada mensagem** - este intervalo de tempo é obtido como um parâmetro de input especificado pelo utilizador quando este arranca o simulador;
 - **Distância máxima entre localizações do sensor entre duas mensagens consecutivas** - este valor é obtido como parâmetro de input especificado pelo utilizador.

Notas sobre o simulador

Este simulador foi pensado e implementado de forma a detetar qualquer falha por parte do utilizador no que diz respeito aos parâmetros por ele inseridos, e dos quais o funcionamento do simulador depende. Deste modo, o simulador contém seguranças como:

- Notificar o utilizador caso exista algum erro de formatação no ficheiro de configuração JSON;
- Notificar o utilizador sempre que algum parâmetro inserido não se encontre no formato esperado, por exemplo, o nome do ficheiro de configuração não estar correto, ou no campo para inserir o nº de mensagens a ser enviadas, o utilizador inserir uma palavra em vez de um valor inteiro;
- Todas as mensagens de erro que o simulador apresenta são elucidativas sobre qual o problema encontrado, levando o utilizador a ir diretamente de encontro ao problema de modo a que o possa resolver de imediato;
- Foi adicionado um comando “- -help”, caso o utilizador tenha dúvidas sobre como deve iniciar o simulador e como deve inserir os parâmetros (nº de mensagens a ser enviado, tempo de intervalo entre elas, etc...);
- Notificar o utilizador quando a ligação ao broker do mapify (a ser explicado mais à frente) é perdida, ou quando não é possível estabelecer a ligação (por exemplo não existir ligação à internet) - esta alteração foi adicionada após a demo realizada à empresa no dia 27 de Abril;
- Apresentar ao utilizador o número de mensagens acumuladas - esta alteração foi adicionada após a demo realizada à empresa no dia 27 de Abril;

Substituição de Valores na Mensagem

O template da mensagem, fornecido pelo ficheiro de configuração, é guardado no formato de string, e a cada geração de uma nova mensagem o programa substitui os campos que são dinâmicos por um valor random dentro dos disponíveis.

No caso do cálculo das coordenadas geográficas do sensor para cada mensagem, já não se trata apenas de uma escolha entre os valores disponíveis para preencher o valor daquele atributo. Caso se trate da 1ª mensagem gerada pelo simulador, o programa guarda os dados fornecidos pelo utilizador no ficheiro de configuração, criando assim uma bounding box (que define os limites geográficos por onde o sensor se pode movimentar) e são geradas as primeiras coordenadas dentro da bounding box, de forma aleatória.

Como já foi referido, o utilizador fornece no arranque do simulador, como parâmetro de input, um STEP máximo, que determina a distância máxima permitida entre localizações desse sensor, entre duas mensagens seguidas. Assim, o simulador irá sempre guardar as coordenadas anteriormente geradas, para

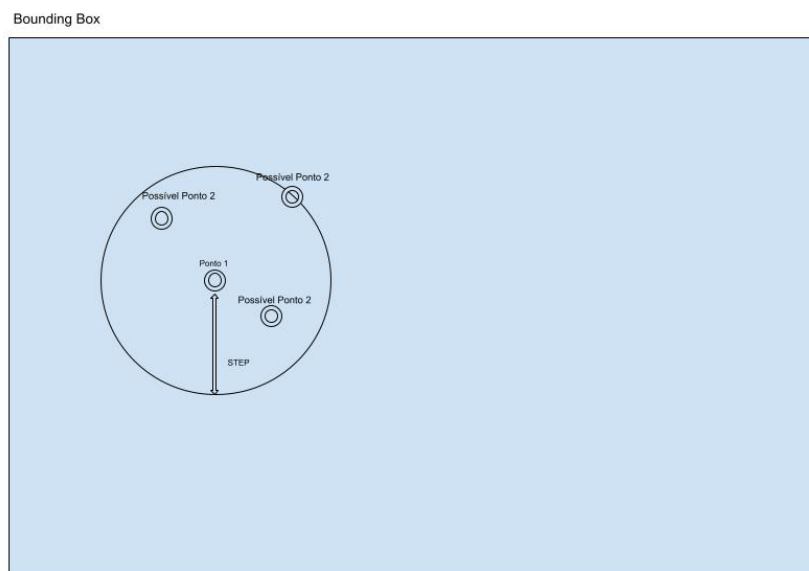
que no cálculo das novas coordenadas, para a nova mensagem, seja possível assegurar-se que este requisito é satisfeito.

De mensagem para mensagem, de localização para localização, as novas coordenadas são geradas da seguinte forma:

- é gerado um ângulo random entre 0 e 360 graus;
- na direção desse ângulo, é criada uma distância random entre 0 e o STEP máximo definido pelo utilizador;
- se o ponto gerado se encontrar dentro da bounding box, esse ponto é aceite e as suas coordenadas são inseridas na mensagem para ser enviada; caso não se encontre dentro da bounding box, o procedimento é manter o ângulo definido e diminuir a distância que foi definida para metade.

Este processo deve repetir-se até que o ponto se enquadre dentro da bounding box.

Exemplo ilustrativo de possíveis localizações de um novo ponto:



Envio das mensagens

Uma vez que todos os campos da mensagem tenham sido tratados e preenchidos, é altura de enviar a mensagem gerada para o Mapify, para que este possa apresentar num mapa, em tempo real, a atividade do sensor e as mudanças dos seus atributos.

Na verdade, a mensagem não é enviada diretamente para o Mapify: esta mensagem é enviada utilizando **MQTT**(Message Queuing Telemetry Transport). **MQTT** é um protocolo de mensagens leve para sensores e pequenos dispositivos móveis otimizado para redes **TCP/IP** (Transmission Control Protocol / Internet Protocol), podendo suportar comunicações assíncronas, ou seja, pode suportar dados transmitidos em fluxos estáveis.

De forma resumida, o processo deste protocolo baseia-se no envio de uma mensagem para um broker, que é capaz de entender e ler múltiplos aparelhos ao mesmo tempo. Mas pode surgir a seguinte questão: o

que é um broker? Uma vez que o esquema de troca de mensagens é fundamentado no modelo **Publisher-Subscriber**, o broker é o elemento responsável por gerir as publicações e as subscrições do protocolo **MQTT**, atuando como um mediador entre as máquinas (neste caso, entre o simulador construído e a plataforma Mapify), permitindo a comunicação entre elas.

Como o simulador pretende enviar várias mensagens com um X tempo de intervalo entre o envio de cada uma delas, foi utilizado o método **setInterval**, suportado por **NodeJS**, que permite realizar um conjunto de ações repetidamente, começando após um determinado período de tempo e repetindo continuamente a cada intervalo. Deste modo, a cada intervalo de tempo definido pelo utilizador, é gerada uma nova mensagem e enviada para o broker do Mapify.

Todo este processo apenas terá fim quando o número de mensagens enviadas atingir o número de mensagens pretendidas pelo utilizador.

Deste modo, é possível correr vários simuladores, ou seja, vários sensores, e a localização destes e os seus atributos serem apresentados num mapa do Mapify, em tempo real.

Sensor Data Visualization Application

Após a conclusão do desenvolvimento do simulador de sensores, foi ainda traçado o objetivo de criar uma pequena aplicação web para que todos os sensores ativos fossem apresentados num mapa em tempo real, à semelhança do que aconteceu na plataforma Mapify, mas desta vez, algo construído por nós.

Este aplicativo foi desenvolvido em **HTML/CSS** e **Javascript** em **NodeJS**, dividindo-se este processo em dois grandes passos:

- criação de um Web Server;
- criação de um objeto cliente (browser).

O Servidor criado, desenvolvido em NodeJS usando Javascript, tem por objetivo servir de intermediário entre o simulador de sensores e a Web Application onde o utilizador poderá ver os sensores dispostos num mapa, em tempo real, como já foi referido.

Inicialmente, o objetivo seria que o simulador enviasse as mensagens por **MQTT** para a plataforma Mapify, e seria esta a responsável por enviar, via **HTTP**, as mensagens provenientes do simulador. No entanto, foi detetado um bug no Mapify que impedia que as mensagens fossem enviadas no formato pretendido, impossibilitando-nos de trabalhar a mensagem recebida no servidor, em que a solução encontrada foi realizar um **POST request** diretamente do simulador de sensores para o servidor web, enviando as mensagens geradas.

Uma vez que era necessário estabelecer ligações, enviar e receber dados, poder-se-ia ter utilizado o módulo **HTTP**, pois permite configurar ligações e receber e enviar dados desde que as conexões utilizem este protocolo. No entanto, considerámos haver uma melhor solução. O framework Express para NodeJS fornece API, sub-módulos, metodologias e convenções para que todos os componentes necessários sejam unidos de forma rápida e simples, para colocar um servidor web funcional. Por este motivo, e por permitir realizar pedidos e envios **HTTP** em menos passos, foi escolhido utilizar o framework Express para se obter o pretendido.

Estabelecida a ligação entre o simulador de sensores e o servidor web, seguiu-se a criação da Web Application. Em **HTML** e **CSS** foi desenvolvida a aplicação gráfica pretendida, utilizando uma **API key** fornecida pelo orientador deste estágio para disponibilizar ao utilizador um mapa da Google Maps.

Aspeto Gráfico da Aplicação:



Na imagem apresentada é possível verificar que foi ativada a opção de serem apresentados ao utilizador no mapa diversos pontos de interesse da região exposta no mapa, utilizando a Maps Javascript API para esse efeito.

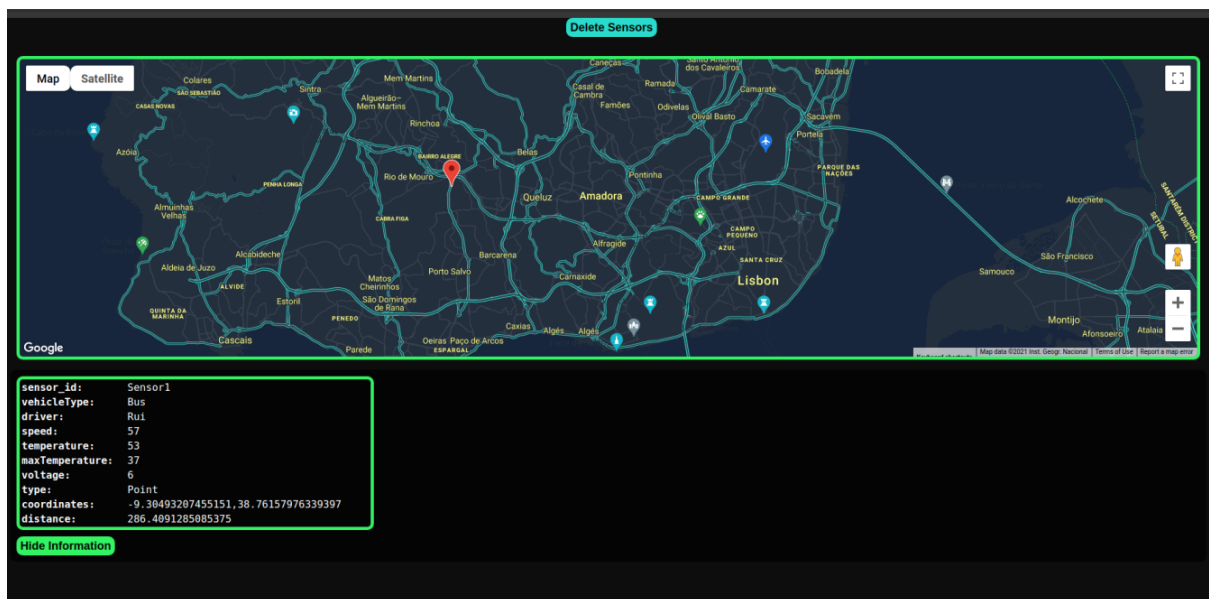
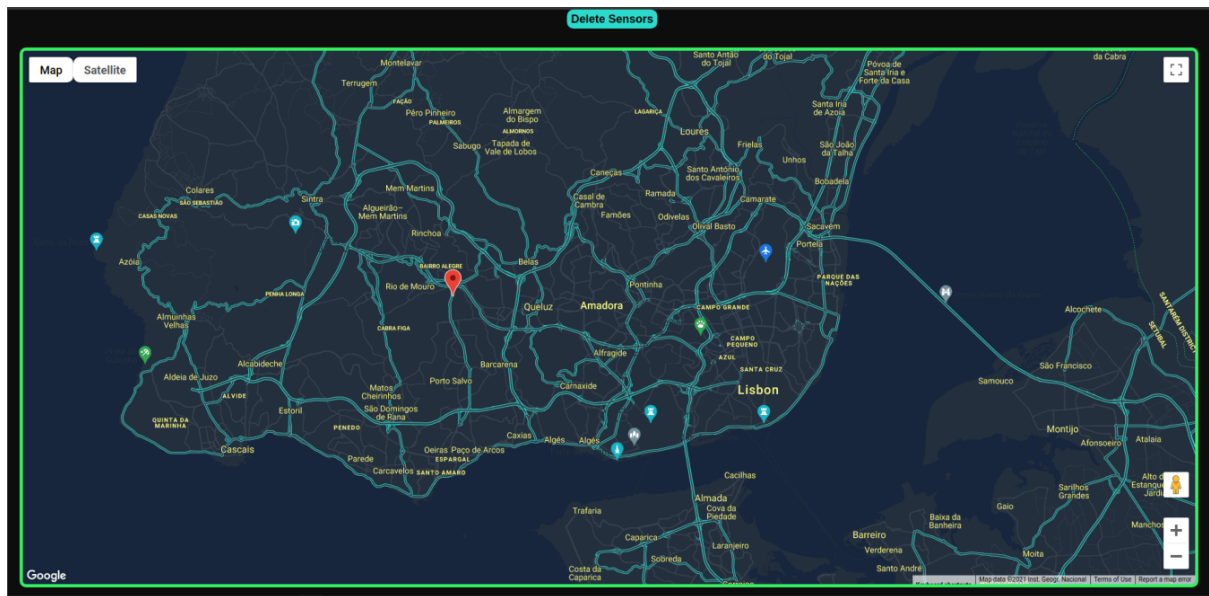
É nesta altura que é desenvolvido o objeto cliente em Javascript. O browser pretende receber informação proveniente do servidor web. Para este efeito, foram utilizados **SSE** (Server-Sent Events), podendo surgir a questão: porque não utilizar **WebSockets**? Ambas são tecnologias que definem como browsers e servidores comunicam entre si. Para este caso, apenas se pretende que a informação navegue do servidor para o browser, e não que esta navegue nos dois sentidos

Enquanto que WebSockets permitem a comunicação **bidirecional** em canais full-duplex, ou seja, comunicação entre servidor-cliente nos dois sentidos, SSE permitem ao cliente receber informação atualizada do servidor através de uma ligação HTTP sem sequer realizar um pedido para a receber, sendo assim esta comunicação **unidirecional** e ideal para a presente situação, pois torna o trabalho com dados em tempo real muito eficiente.

Mas e se existir mais que um cliente ligado ao servidor? Para que possam haver vários clientes conectados, que pretendem receber as mensagens emitidas pelo simulador de sensores, na implementação do servidor foi utilizada uma **UUIDV4** (Universally Unique Identifier Version 4) **Generator Tool**. Esta ferramenta, usando números random, permite gerar um identificador único universal, que distingue cada utilizador.

Assim, a cada novo utilizador, este é adicionado a uma lista existente no servidor, e sempre que sejam recebidas mensagens do simulador de sensores, o servidor irá enviar as novas mensagens, via SSE, para todos os utilizadores subscritos. Desta forma, cada utilizador poderá ver no seu mapa, a atualização da localização dos sensores e os seus atributos em tempo real.

Resultado Final da Web Application:



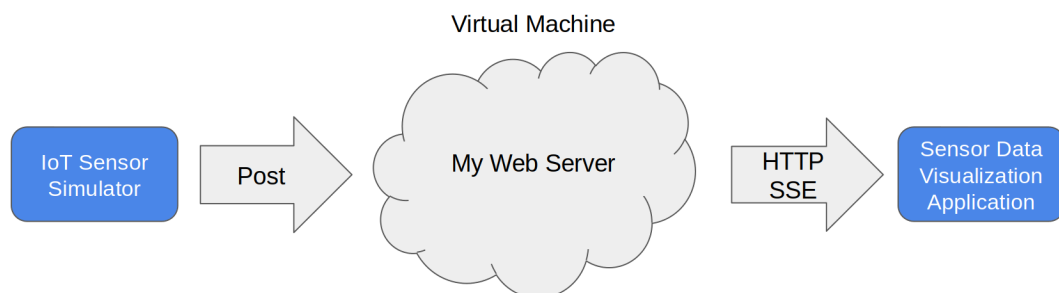
Como é possível observar nas duas imagens aqui apresentadas, um sensor é representado no mapa por um **Marker**. Esta parte da implementação da aplicação tem de ser realizada em client-side, para que o browser saiba quais os sensores que já tem presentes no mapa e quais ainda não tem, para quando este recebe a informação proveniente do servidor, saber se deve criar um novo Marker, ou atualizar um. Assim, para cada cliente, todos os novos sensores são guardados num array, identificados pelo seu ID, e quando é recebida informação é verificado se o sensor já existe ou não. Se existir, este é atualizado; se for um novo sensor, é criado um novo Marker e adicionado à lista de sensores. Para que isto resulte, sempre que o botão "Delete Sensors" é premido (explicado adiante), todos os sensores já inativos são eliminados da lista de sensores do cliente.

Para terminar a descrição detalhada deste projeto, resta apenas referir algumas funcionalidades da Web Application desenvolvida, tais como:

- Ao passar o cursor por cima de um Marker no mapa, surge para o utilizador o ID do sensor representado;
- Se o utilizador clicar num sensor, ser-lhe-á apresentada uma tabela com os valores atualizados de todos os atributos daquele sensor, correspondendo à última mensagem recebida pelo browser sobre aquele sensor;
- Foi criado um botão para que o utilizador possa fazer desaparecer a tabela informativa e voltar ao modo em que o mapa ocupa quase toda a página Web;
- Se o utilizador estiver em modo Full Screen e pretender observar as especificações de um determinado sensor, basta clicar neste e a página sai automaticamente do modo Full Screen, apresentando a tabela;
- Foi criado também um botão "Delete Sensors" para situações em que o utilizador queira "limpar" o seu mapa, por exemplo: o utilizador tem dez sensores dispostos no mapa, no entanto, oito deles já terminaram a sua execução; premindo o botão "Delete Sensors", o utilizador elimina imediatamente do seu mapa todos os sensores, no entanto, aqueles que ainda se encontravam ativos, aparecerão de novo no mapa quando for recebida nova informação sobre os mesmos. Desta forma o utilizador pode observar apenas os sensores que ainda se encontram ativos.

Por fim, de modo a testar todos os aplicativos e as suas ligações sem ser localmente, o servidor foi inserido e executado numa VM de **Google Cloud**, tendo sido atingido mais um objetivo deste estágio.

Esquema elucidativo do funcionamento do projeto final:



5 Avaliação crítica

Chegado a esta fase do relatório, importa refletir acerca das minhas expectativas para este estágio, se foram ou não atingidas, e os aspetos que considero mais relevantes, que de alguma forma irão influenciar a minha vida profissional.

Confesso que inicialmente senti alguma frustração pelo facto de ser um estágio integralmente à distância. Uma das minhas expectativas correspondia mesmo à forma de como me iria integrar numa equipa, numa empresa e no mercado de trabalho. No entanto, devido ao contexto pandémico atual, este estágio, tal como as nossas vidas atualmente, teve que ser adaptado a esta realidade.

Possivelmente, o teletrabalho será um método de trabalho cada vez mais adotado no futuro, em algumas áreas, inclusive a de Engenharia Informática.

No que diz respeito ao tipo de formação em contexto laboral, é importante referir que consegui realizar o meu estágio na empresa que seleccionei como primeira opção, pois logo na apresentação dos campos de estágio aos alunos, fiquei interessado no tipo de estágio que a empresa oferecia, e nos projetos em que se encontra envolvida.

A receção por parte do orientador de estágio da empresa e dos restantes elementos da equipa, não poderia ter sido melhor. Talvez pela metodologia de trabalho utilizada na Focus BC e pelo ambiente informal que se vivencia entre todos os colaboradores, independentemente do vínculo e lugar que ocupam na hierarquia da mesma, rapidamente me senti integrado. Por vezes este ambiente é mais vantajoso no desenvolvimento das nossas tarefas, tornando-nos mais produtivos e criativos. O facto de ser à distância em regime de teletrabalho, levou-me a pensar que isso talvez fosse um obstáculo para a minha integração e aprendizagem; no entanto, tal não se verificou. Embora continue a defender que o regime presencial tem outras vantagens, principalmente quando estamos a tentar absorver toda a informação circundante.

Criar algo em equipa, em que os elementos se complementam foi um dos aspetos que considerei importantes e facilitador na integração da mesma. A utilização de algumas ferramentas e programas, nomeadamente: Typescript em NodeJs, no desenvolvimento do simulador de sensores e na sua aplicabilidade prática na vida real, enquanto produto que está a ser criado de acordo com os requisitos e necessidades do cliente, agradou-me muito. O facto de ter feito o planeamento e o desenvolvimento de software em ambiente Agile, também me satisfaz bastante e proporcionou-me conhecimentos que eventualmente irei utilizar de futuro, já em contexto laboral, visto que é neste ambiente que a maioria das empresas desta área trabalham.

Para concluir, julgo ser este o momento de mencionar que estas 14 semanas foram uma mais valia no meu processo de aprendizagem, espaço temporal em que senti que evolui imenso na minha formação académica.

No entanto, penso que também é este o momento em que devo referir que seria ainda mais proveitoso se fosse um estágio realizado em tempo integral, cinco dias por semana, em que apenas nos dedicávamos a este tipo de formação em contexto de trabalho. Conciliar formação em contexto académico com a formação em contexto laboral em simultâneo, julgo que apenas contribuiu para um dispersar de atenção e de energia, em que não conseguia dedicar-me em pleno a um ou a outro tipo de formação. Deixo aqui o meu testemunho, que poderá eventualmente servir de sugestão para o futuro, em que defendo que este tipo de formação (em contexto laboral) deveria ser realizado em exclusivo, nem que para isso o número de semanas de estágio tivesse que ser reduzido.

Por tudo o que já referi anteriormente, e após uma reflexão consciente, julgo que realizei o estágio de uma forma muito positiva, em que consegui superar algumas dificuldades técnicas e que sem dúvida me enriqueceu bastante a nível de conhecimentos.

Referências

- [1] Focus BC. <https://www.focus-bc.com/>.
- [2] Mapify. <https://www.mapify.ai/>.
- [3] MQTT. <https://www.npmjs.com/package/mqtt>.
- [4] Express. <https://www.npmjs.com/package/express>.
- [5] Maps JavaScript API. <https://developers.google.com/maps/documentation/javascript/overview>.
- [6] Markers. <https://developers.google.com/maps/documentation/javascript/markers>.