

Desenvolvimento de um sistema de monitorização e controlo de estacionamentos públicos por meio da Internet

Wesley Victor da Costa Vieira¹ and Pedro Guimarães Marçal¹

¹Departamento de Física e Astronomia, Faculdade de Ciências da Universidade do Porto

26 de junho de 2021

Resumo

O presente relatório detalha o processo de desenvolvimento ao nível da unidade curricular de Laboratório de Aquisição e Controlo de um sistema de monitorização e controlo de vagas de estacionamentos públicos através da Internet. Em termos de aquisição de dados, foi desenvolvido um sensor que detecta se uma vaga de um estacionamento está livre ou ocupada usando um Arduino Uno. Desenvolveu-se também uma versão básica dos principais componentes do sistema apresentado. Utilizou-se o programa LabVIEW para fazer a interacção entre os sensores (que denominamos por *nodes*) e o servidor web. A informação no servidor foi então apresentada numa interface web onde utilizadores podem visualizá-la em tempo real. São discutidas ainda as implicações de segurança que uma arquitectura deste tipo requer e algumas soluções para o problema da centralização de dados num servidor central.

1 Introdução

A maioria dos seres humanos actualmente vive em áreas urbanas e é expectável que cerca de dois terços da população mundial irá viver em cidades até 2050 [1]. Neste sentido, o consequente aumento da densidade populacional em grandes cidades acarreta o aumento da utilização de veículos automóveis. Levando estes factos em consideração, entender os padrões de mobilidade urbana torna-se extremamente importante para a redução de gases do efeito estufa emitido por veículos com motores de combustão assim como no desenvolvimento de infraestruturas públicas e políticas de mobilidade sustentáveis.

Tendo isto em vista, propõe-se neste trabalho o desenvolvimento de um sistema que permite a monitorização e controlo do estacionamento de veículos na via pública por parte das cidades, ao mesmo tempo oferecendo uma plataforma tecnológicas para localização de vagas livres nas zonas da cidade aos motoristas.

Alguns benefícios de se implementar um sistema deste tipo incluem:

- Produção de dados para análise por parte das cidades para melhoramento de infraestruturas públicas e planeamento
- Redução do tempo de procura por vagas de estacionamentos.
- Redução de emissão de CO₂ emitidos por veículos movidos por motores de combustão.

Este trabalho focou-se primariamente no desenvolvimento de um protótipo de aquisição e processamento de dados juntamente com uma plataforma web simples que simula uma versão reduzida da arquitectura aqui detalhada. Os demais componentes do sistema proposto, ainda que não implementados por completo, serão discutidos com relativo detalhe. A próxima secção detalha quais foram os componentes electrónicos utilizados para o desenvolvimento do protótipo. Na secção 3, é detalhado o funcionamento do sistema e as diversas partes que o compõem. Na secção 4, é apresentado aquilo que foi realizado nas aulas laboratoriais e os resultados obtidos ao nível das implementações e funcionalidades desejadas para o projecto.

1.1 Caso de estudo - Baixa do Porto

Neste momento existem mais de 12.000 lugares de estacionamento públicos pagos na cidade do Porto. Encontram-se divididos em quatro diferentes zonas, cada uma com diferentes tarifas [2]. A fiscalização deste sistema é feito por diferentes empresas contratadas pela Câmara do Porto. Esta rede de lugares de estacionamento produz entre 10.000€ a 100.000€ por dia, excluindo a receita proveniente de multas. A introdução de um sistema de gestão desta rede de estacionamentos teria como objectivos principais o ajuste e controlo das tarifas de diferentes zonas assim como permitir à Câmara do Porto saber quais as zonas mais utilizadas, a que horas e deste modo melhorar a rede de transportes da cidade de modo a servir os seus habitantes. Os autores deste trabalho acreditam que será uma questão de tempo até que a infraestrutura pública de parquímetros esteja auxiliada por detectores de presença de veículos similares ao apresentado neste relatório.

2 Levantamento de requisitos

Começou-se por delimitar quais seriam os equipamentos utilizados para atingir os objectivos do projeto. Inicialmente foi considerada a opção de utilizar uma DAQ (abreviação de *Data Aquisition Systems*) para fazer aquisição dos dados do sensor, entretanto optou-se por utilizar um Arduino Uno, apresentado na figura 1, para lidar tanto com a aquisição como com o processamento dos dados vindos do sensor. Entre os motivos para esta escolha estavam o facto de ser um sistema open source - sendo que em caso de dúvidas existe um conjunto vasto de materiais e documentação disponível online - e, talvez mais importante, o facto de que era desejável uma separação entre os componentes do sistema. Como será mencionado na secção 4, o Arduino e o programa LabVIEW estão em níveis lógicos separados, sendo que o Arduino foi responsável pela aquisição dos sinais do sensor e pelo algoritmo que determina o estado de uma vaga enquanto que o programa em LabVIEW ficou responsável pela comunicação deste estado com um servidor central. Mais detalhes sobre os componentes e arquitectura do sistema serão dados na secção seguinte.

A ideia original era utilizar um magnetómetro (modelo HMC5883L) para medir as componentes do campo magnético ao redor da vaga de estacionamento. Quando um veículo se aproxima do sensor, estes valores mudam e é possível determinar dessa forma se uma vaga está ocupada ou livre. Entretanto, devido a motivos técnicos desconhecidos pelos autores do trabalho, o sensor deixou de funcionar no decorrer do desenvolvimento do protótipo, havendo apenas um exemplar foi necessária a escolha de um outro sensor.

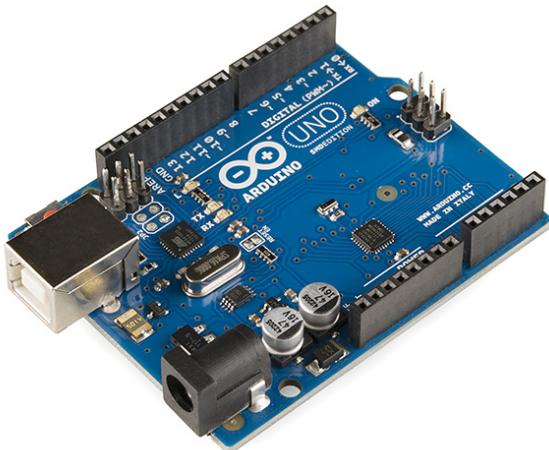


Figura 1: Modelo Arduino UNO utilizado para a aquisição e processamento de dados de sensor um sensor de presença óptico.

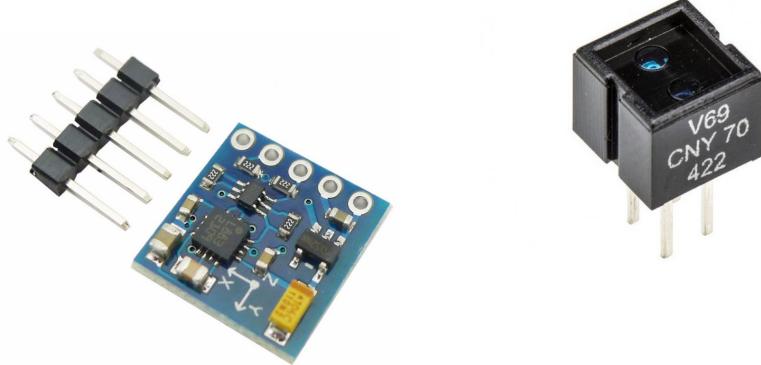


Figura 2: **Esquerda:** Sensor Magnetómetro HMC5883L. **Direita:** Sensor óptico reflectivo CNY70.

Após consideração, e com ajuda do professor da disciplina, decidiu-se utilizar um sensor óptico de presença (Figura 2) como substituto ao magnetómetro. É bem verdade que em uma situação real um sensor óptico de presença não seria a escolha ideal (qualquer objecto que se aproximasse do sensor poderia facilmente alterar o seu estado). Entretanto, o sensor óptico é sim útil como uma simples alavanca booleana para demonstrar o conceito do sistema.

3 Arquitectura do Sistema

Antes de detalhar a função de cada componente do sistema, é necessário introduzir a nomenclatura dada a cada uma das partes que o compõem. A figura 3 apresenta um esquema geral de como funcionaria o sistema de monitorização e controlo de estacionamentos públicos em um cenário real.

Denomina-se por *nodes* os sensores que detectam o estado de uma vaga de estacionamento. Os *hubs* recebem informações de diversos *nodes* e as transmite para um servidor web central. No servidor encontram-se o estado de todos os *nodes* do sistema actualizados em tempo real. Estes dados podem então ser apresentados numa interface web em que utilizadores, através de uma aplicação no telemóvel por exemplo, conseguem interagir com o intuito final de achar vagas de estacionamento livre mais próximos de si.

Na secção 4 será detalhado o processo de desenvolvimento do protótipo que funcionará como ”proof of concept” da ideia em cada um dos níveis apresentados nesta secção.

3.1 Nodes

Os sensores presentes nos estacionamentos da via pública constituem a base do sistema de monitorização das vagas de estacionamento. São eles que determinam o estado da vaga, i.e se existe um veículo estacionado ou não numa vaga de estacionamento. Além do estado de ocupação da vaga, os *nodes* devem também possuir um identificador único para que seja possível associar a eles informações de geo-localização, mais especificamente os valores da latitude e longitude do sítio em que o sensor, e por consequência a vaga, se encontra.

Devido a sua importância no sistema, uma atenção particular será dada a sua implementação e desenvolvimento na secção 4.1.

3.2 Hubs

Os *hubs* tem como principal função agregar estados de várias vagas de estacionamento de uma determinada zona e servir como ponte de comunicação entre *nodes* e o servidor central. *Hubs* devem possuir informações sobre a zona em que fazem parte e, assim como os *nodes*, também devem possuir um

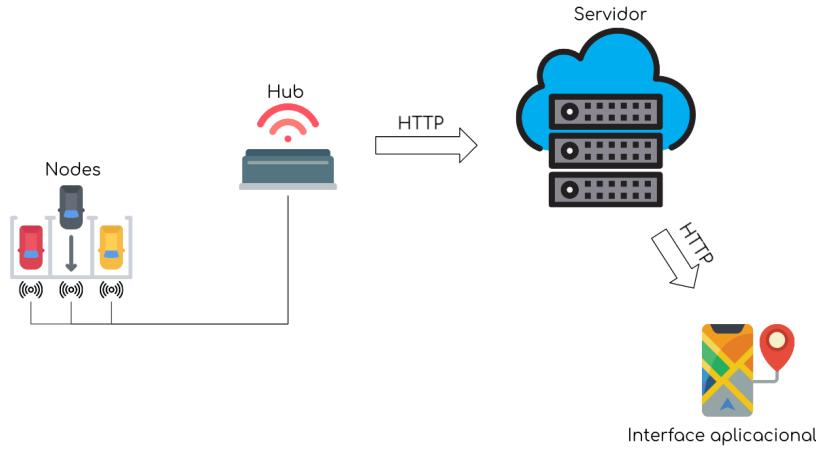


Figura 3: Arquitectura do sistema dividida em nodes, hubs, servidor web e interface aplicacional.

identificador único. Os *hubs* devem possuir uma placa de rede para poderem enviar (e possivelmente receber) informações por meio da Internet. Este requisito é particularmente importante para que os estados dos *nodes* sejam actualizados em tempo real no servidor central.

É desconhecido pelos autores do trabalho as especificidades dos parquímetros actualmente usados para o pagamento das vagas de estacionamento, entretanto já existem serviços de pagamento de estacionamentos online e poderia-se utilizar da estrutura já estabelecida para conectar *nodes* de uma determinada zona aos seus respectivos *hubs*.

3.3 Servidor

O servidor é a componente que armazena o estado de todas as vagas activas no sistema. É composto por um servidor web que recebe e envia os estados das vagas utilizando-se do protocolo aplicacional da Internet, o HTTP (Hypertext Transfer Protocol), assim como uma base de dados onde são armazenadas os estados de todos os *nodes*. Para que a comunicação entre os *hubs* e o servidor funcione, é necessária a criação de uma API (Application Programming Interface) que possui como principal objectivo servir de interação entre sistemas e possibilitar operações de coleta e alteração estados de *nodes* no servidor central usando os protocolos da Internet.

Um ponto importante, e que será de certa forma ignorado no desenvolvimento do protótipo, é a segurança da comunicação entre hubs e servidor. Como esta comunicação é feita por meio de pedidos HTTP, é necessário haver um conjunto de medidas de segurança que garantem que, por exemplo, quem está a efectuar o pedido para alterar o estado de um determinado node é realmente um hub válido, e não um agente externo que está a tentar se beneficiar do sistema. Outros detalhes sobre este assunto são mencionados na secção 3.5.

3.4 Interface do utilizador

Os utilizadores do sistema interagem com os dados por meio de uma plataforma web. É nela que os estados de cada *node* são apresentados para os utilizadores.

Ultimamente, seria possível realizar a integração com um serviço web de mapas para conectar motoristas e as vagas do sistema numa aplicação utilizando o sistema de GPS dos telemóveis. Dessa forma, os utilizadores poderiam utilizar todas as funcionalidades de um aplicativo de mapas - como criação de rotas, cálculo de distâncias e tempo, entre outras - para achar e ir em direcção a vagas livres mais próximas do sítio onde estão.

3.5 Considerações de segurança

Levando em conta que o sistema armazena informações sensitivas num servidor central, esta arquitectura está sujeita a diversos vectores de ataque, entre eles ataques DDoS (Distributed Denial of Service), hacking, e outros que podem colocar a integridade das informações de sistemas e utilizadores

em risco. É de notar que a implementação discutida na secção seguinte não leva nenhum desses pontos em consideração, focando-se simplesmente na implementação das funcionalidades básicas do sistema.

Uma alternativa a esta arquitectura, que entretanto não será explorada no presente trabalho, seria um sistema distribuído que não requer a presença de uma autoridade central (neste caso um servidor controlado por uma entidade). Uma tecnologia como Blockchain [3] poderia ser usada para descentralizar a estrutura do sistema, além de providenciar uma camada adicional de segurança uma vez que a Blockchain utiliza-se de criptografia para garantir a fidedignidade das transacções entre os nós da rede sem a necessidade de confiança entre eles.

Outro problema que a implementação de um sistema destes teria seria a danificação e respectiva manutenção da rede física. Tanto os hubs como os sensores em si teriam de ser feitos de modo a minimizar as suas outras potenciais utilizações, assim diminuindo a probabilidade de serem roubados ou destruídos.

4 Protótipo

De facto a preparação ou montagem de uma rede de sensores não é exequível no plano de uma unidade curricular nem haveria material disponível para tal. Com isto, decidiu-se que a melhor solução seria criar um programa no LabVIEW que simule o estado de vários nodes de estacionamento e teríamos apenas um sensor físico o qual serviria de protótipo experimental. Foi então construído um algoritmo com base na geração de números aleatórios que simula o processo de ocupação e desocupação de lugares de estacionamento.

4.1 Sensor

Conforme foi mencionado anteriormente, não foi possível utilizar um sensor magnetómetro para o desenvolvimento do protótipo como se havia planeado. Entretanto, o protótipo foi planeado de tal forma que o algoritmo de detecção do estado da vaga retorna, seja utilizado um sensor de presença ou um magnetómetro, o mesmo resultado. É então escrito na porta USB do computador um valor booleano que representa o estado da vaga - 1 representa uma vaga ocupada e 0 uma vaga livre. Dessa forma, para fazer com que o protótipo fosse mais fidedigno a um cenário real com a utilização de um magnetómetro, bastaria alterar apenas o algoritmo (implementado no Arduino) que retorna o estado actual de uma vaga de estacionamento. A implementação deste algoritmo fica, portanto, encapsulada do mundo exterior e vem a ser útil caso mudanças na implementação dos *nodes* venham a ser feitas. Fica portanto o algoritmo que detecta ou não a presença no Arduino, deixando para o LabVIEW apenas um sinal booleano. A figura 4 mostra o circuito utilizado para a montagem e a figura 5 apresenta o resultado final do protótipo que realiza a função de um *node* na arquitectura proposta. Nesta última, verifica-se que foi usado 2 LEDs para representar visualmente qual o estado da vaga.

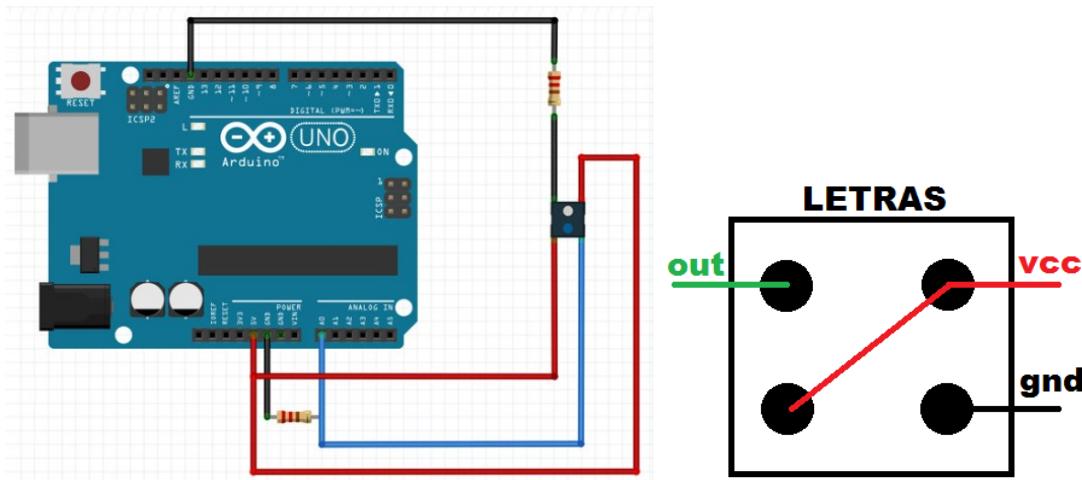


Figura 4: Esquerda - circuito utilizado no protótipo com resistências de $1\text{k}\Omega$ (canto inferior esquerdo) e 200Ω (no canto superior direito) . Direita - especificações do sensor óptico.

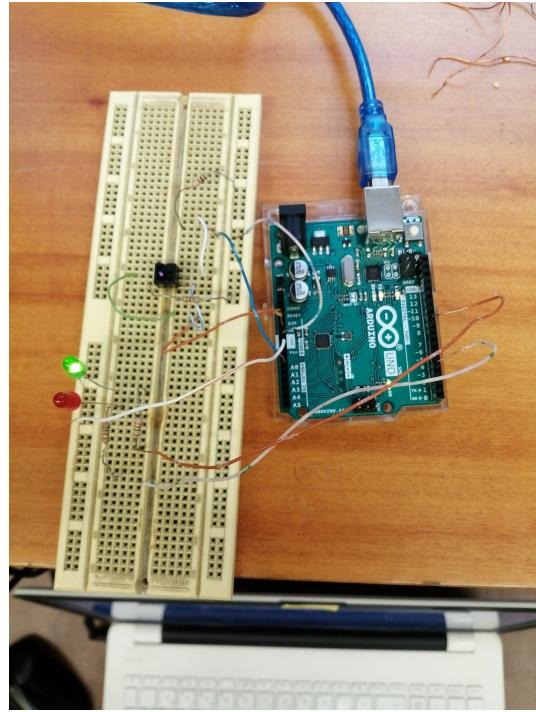


Figura 5: Implementação final do protótipo de um *node* simples.

4.2 LabVIEW

Para demonstrar como um sistema desse tipo funcionaria com vários nodes, desenvolveu-se um programa em LabVIEW que gera dados aleatórios com o intuito de simular vagas de estacionamento (figura 6). A zona simulada, cujo o node representado pelo protótipo desenvolvido faz parte, é então uma de várias outras vagas "virtuais" simuladas no LabVIEW, sendo que é possível configurar, entre outros parâmetros, a periodicidade com que as vagas são ocupadas - podendo por exemplo simular a evolução da ocupação das vagas de estacionamento em horários pico num dia comercial. Implementou-se um VI (figura 7) com uma interface que apresenta os dados dos *nodes*. De modo a simplificar a apresentação do interface, adicionou-se legendas das diferentes aplicações do VI. Considerou-se que uma das maneiras mais simples de representar zonas diferentes seria com Tabs juntamente com a utilização de LEDs e valores booleanos para representar os lugares de estacionamento.

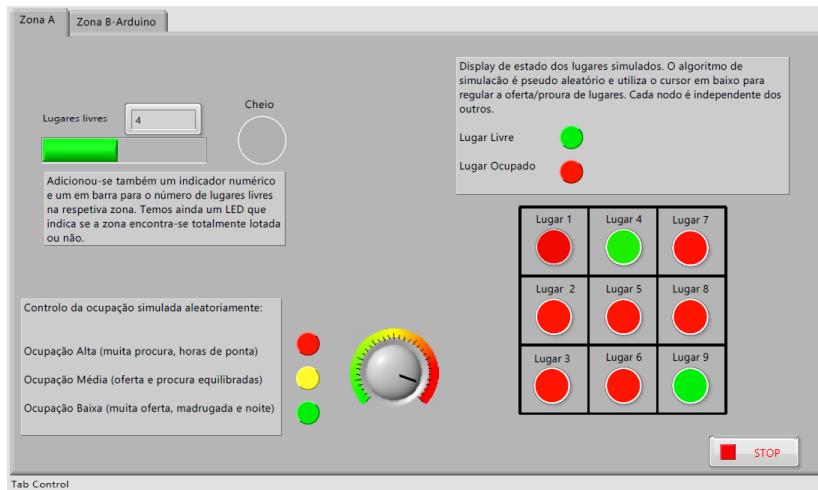


Figura 6: VI desenvolvido para simular dados aleatórios dos estados das vagas de estacionamentos.

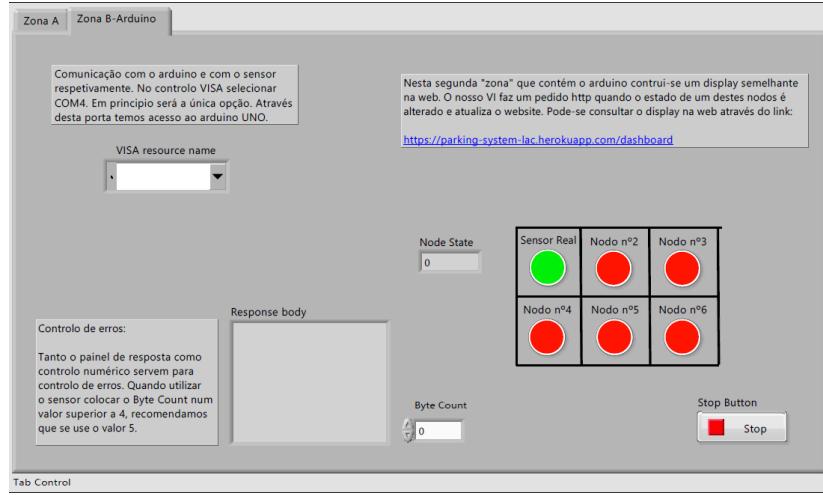


Figura 7: VI com os dados do sensor e com informações relativas aos pedidos HTTP e os valores lidos da porta USB.

4.3 Comunicação entre o LabVIEW e o servidor web

Em termos de componentes, o computador utilizado para executar o programa LabVIEW funciona essencialmente como um *hub*, fazendo a interacção entre o sensor e o servidor web. O programa então lê da porta USB o estado da vaga e realiza pedidos HTTP para o servidor, actualizando o estado da vaga toda vez que há uma mudança (livre passa para ocupado e vice-versa). A figura 8 apresenta o programa em LabVIEW utilizado para fazer pedidos HTTP ao servidor web.

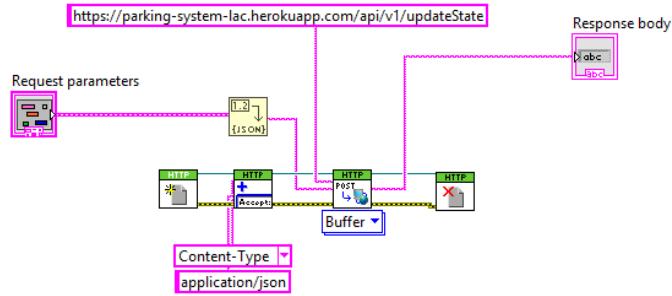


Figura 8: SubVI que realiza pedidos HTTP.

4.4 Controlo de erros

De modo a controlar ou a limitar os erros de leitura por parte do sensor, ou pelo menos limitar a passagem de sinais de mudança de estado do nodo para a interface, adicionou-se ao algoritmo de detecção do estado de um estacionamento uma parte que lida com valores iniciais devolvidos pelo sensor óptico. Este algoritmo guarda os valores de luminosidade quando se liga o sensor e assim tem uma base a partir do qual compara os valores obtido para a presença ou não de um objecto na sua proximidade. Deste modo conseguimos não só evitar ter de calibrar o sensor para diferentes luminosidades ambientes mas também conseguimos melhor precisão na detecção de um obstáculo.

4.5 Servidor Web

As informações relativas aos sensores (códigos identificadores, latitude e longitude) foram armazenadas numa instância da base de dados relacional MySQL, conforme mostra a figura 9. Ao receber um pedido HTTP de alteração do estado de um *node*, a informação relativa a esta vaga em específico é então actualizada no servidor, sendo então possível que utilizadores vejam os estados das vagas em tempo real assim como também a suas localizações individuais. Esta alteração é feita através de pedidos HTTP com o método POST feitos pelo programa LabVIEW. Abaixo encontra-se um exemplo do corpo de um pedido, formatado em JSON (JavaScript Object Notation):

```
{
    "idNode": "A1",
    "state": 0
}
```

Os pedidos HTTP, nesta implementação simplificada do sistema, apenas levam no corpo qual o id do *node* cujo estado deve ser alterado e qual o novo estado da vaga. Conforme já foi discutido, nenhuma medida de segurança foi adoptada na interacção entre os *hubs* e o servidor por motivos de simplificação. Isto constituiria, como seria de se imaginar, uma vulnerabilidade gigantesca caso esta implementação fosse utilizada em um cenário real e seria necessário dar atenção a questões de autenticação e segurança na comunicação entre os componentes do sistema. Usando a API desenvolvida ¹ é possível alterar os estados das vagas por meio de pedidos HTTP. Nota-se também na figura 9 que os identificadores dos *nodes* são relativamente simples, entretanto numa implementação real teria de ser dado atenção especial a estes valores uma vez que eles devem ser únicos para cada *node*. Ao receber estas informações, o servidor actualiza o estado da vaga e a disponibiliza na plataforma web juntamente com informações de todos os outros *nodes* cadastrados.

Vale a pena notar que a aplicação web do actual protótipo apenas retorna o estado de cada node registado. Numa iteração futura, seria possível implementar o sistema com a API do Google Maps, por exemplo, para que utilizadores conseguissem ver já num mapa precisamente o local que se encontram as vagas de estacionamento na cidade do Porto (juntamente com os seus estados em tempo real).

id	nodeId	state	latitude	longitude
1	A1	0	41.152830	-8.637318
2	A2	1	41.152819	-8.637258
3	A3	0	41.152806	-8.637199
4	A4	1	41.152792	-8.637131
5	A5	0	41.152777	-8.637075
6	A6	1	41.152734	-8.636884

Figura 9: **Esquerda:** Dados referentes aos *nodes* numa base de dados relacional. **Direita:** Página web visualizada num dispositivo móvel que apresenta as informações em tempo real de cada node do sistema, actualizando automaticamente a cada mudança de estado.

¹O código utilizado para implementar o servidor web está disponível em github.com/w3slley/parking-system-lac

5 Conclusões

No presente trabalho desenvolveu-se uma arquitectura simples para um sistema de monitorização e controlo de vagas de estacionamentos públicos, tendo sido implementado as funcionalidades fundamentais de cada componente do sistema. No contexto da disciplina, foi necessário o conhecimento de LabVIEW e Arduino para a implementação de algumas funcionalidades, tanto a nível de aquisição dos dados quanto ao processamento destes. Uma atenção especial foi dada à interacção entre esses dois sistemas. Um seguimento natural a este projecto seria utilizar de facto um sensor do campo magnético para determinar o estado de lugares reais, assim como modelar um programa que permitisse a um sensor simples (por exemplo aquele que seria usado neste trabalho) medir com precisão a presença ou não de um veículo. Finalmente, um sistema como este poderia ser implementado a nível de teste em zonas de estacionamento altamente utilizadas de uma cidade como o Porto. Conforme também foi brevemente mencionado, para solucionar o problema do ponto de falha central do sistema, tecnologias como Blockchain poderiam ser utilizadas para criar um sistema seguro, distribuído e descentralizado.

Referências

- [1] Department of Economic United Nations and Social Affairs. Population division world urbanization prospects: The 2014 revision, highlights. 2014. <https://www.un.org/en/development/desa/publications/2014-revision-world-urbanization-prospects.html>.
- [2] Mobilidade Câmara Municipal do Porto. Zonas com estacionamento pago. 2020. <https://mobilidade.cm-porto.pt/estacionamento-na-via-publica/estacionamento-1>.
- [3] Weizhi Meng, Jianfeng Wang, Xianmin Wang, Joseph Liu, Zuoxia Yu, Jin Li, Yongjun Zhao, and Sherman S. M. Chow. Position paper on blockchain technology: Smart contract and applications. 2018.