

## E4 - Descrição de tarefas e entregáveis

### Lista de Tarefas

#### 1. Sensores (Ricardo Carvalho e Rodrigo Rosmaninho):

- 1.1. Investigar e Decidir os modelos de sensores e tecnologias de comunicação a utilizar (RC e RR) : Consultar as especificações e manuais dos sensores existentes no MakerLab e no Instituto de Telecomunicações, assim como tecnologias de comunicação dos dados provenientes dos sensores, e fazer uma seleção do material que irá ser utilizado na implementação.
- 1.2. Planear as instalações nas salas prova de conceito (RC): Fazer um plano/esquema das instalações que se irão realizar na implementação.
- 1.3. Programar / Instalar módulos de sensorização nas salas prova de conceito (RC e RR): Instalar os sensores e configurar estes de modo a poder ser possível a “extração” de dados destes sensores. Fazer a instalação destes sensores nas salas selecionadas para a prova de conceito.
- 1.4. Programar / Instalar Gateway de informação de sensores (RR): Configurar o servidor que irá receber e encaminhar os dados de sensorização para a plataforma IoT (ie.: Eclipse Hono) e instalar o mesmo onde haja conexão com a rede.
- 1.5. Integrar informação de sensorização na API (RC e RR): Ajudar na implementação e teste dos endpoints associados ao acesso à informação recolhida pelos dispositivos de sensorização.
- 1.6. Documentar a arquitetura, o código elaborado e as instalações realizadas (RC e RR): Documentar em relatórios bem estruturados o planeamento, o código efectuado na programação dos sensores e o resultado das instalações realizadas.

#### 2. API (Eurico Dias e Gonçalo Perna):

- 2.1. Definir arquitetura lógica (ED e GP): definir a organização estrutural e funcional da API e o fluxo de funcionamento entre as suas sub-componentes lógicas.
- 2.2. Investigar serviços, frameworks e métodos de autenticação (ED e GP): investigar e escolher, com base na arquitetura lógica definida, as tecnologias mais apropriadas a utilizar para desenvolver ou suportar as componentes lógicas da API.
- 2.3. Decidir método de persistência de dados a usar (ED e GP): decidir, de acordo com a arquitetura lógica desenhada, qual o paradigma e que tecnologias de persistência de dados serão utilizadas.

- 2.4. Definir modelo de base de dados (ED e GP): desenhar o diagrama ou conjunto de diagramas (ER, UML, etc.) que permitem definir de forma inequívoca o modelo de base de dados/persistência a usar.
- 2.5. Definir arquitetura de componentes da API (ED e GP): definir e desenhar a arquitetura de componentes da API.
- 2.6. Instalar e configurar sistema de base de dados (ED): instalar e configurar o sistema de persistência de dados escolhido.
- 2.7. Definir endpoint de teste da API (GP): desenvolver e testar um endpoint que permita exemplificar, de forma rápida, o funcionamento e fluxo de dados da arquitetura definida.
- 2.8. Implementar endpoints de autenticação (ED): implementar e testar os endpoints relativos ao método de autenticação escolhido.
- 2.9. Criar endpoints de acesso à informação de sensorização (ED e GP): implementar e testar todos os endpoints associados ao acesso à informação recolhida pelos dispositivos de sensorização.
- 2.10. Programar endpoints de acesso a informação de natureza variada (ED e GP): criar e testar todos os outros endpoints para recolha ou inserção de informação que não é proveniente dos dispositivos de sensorização.
- 2.11. Documentar código, arquitetura e configurações (ED e GP): documentar processos, fluxos e outputs de definição do código desenvolvido, vistas de arquitetura definidas e configurações de tecnologias e ferramentas instaladas e/ou utilizadas para suportar o desenvolvimento da API.

### **3. Frontend (João Trindade e Pedro Valério):**

- 3.1. Investigar frameworks para CSS (JT): Procurar e definir frameworks de css que melhor se enquadrem no projecto
- 3.2. Investigar frameworks para o desenvolvimento de aplicações móveis (PV): Procurar e definir frameworks de desenvolvimento de aplicações móveis relevantes e úteis para o desenvolvimento da aplicação, de acordo com a arquitectura definida
- 3.3. Investigar frameworks para JavaScript (PV): Procurar e definir frameworks de javascript relevantes e úteis para o desenvolvimento do dashboard, de acordo com a arquitectura definida
- 3.4. Criar Dashboard (JT e PV): Implementar uma dashboard onde os dados recolhidos pelos vários sensores e fornecidos pela API possam ser visualizados
  - 3.4.1. Testar comunicação com API (JT e PV): Garantir a correcta comunicação com a API, através de testes simples e representativos, terem, necessariamente dados relevantes

- 3.4.2. Fazer o layout da Dashboard (JT): Desenhar um layout para o dashboard, considerando a usabilidade, interatividade e fácil compreensão do mesmo
- 3.5. Criar Aplicação Móvel (JT e PV): Implementar uma aplicação móvel de fácil uso que permita obter informações relevantes sobre cada sala, tal como a sua utilização
  - 3.5.1. Fazer o layout da Aplicação (JT): Definir um layout para a aplicação móvel que possibilite uma fácil utilização

## **Milestones**

**M1** - Apresentação dos objetivos e calendarização do projeto: marca a apresentação do MVP para o sistema a desenvolver, com a calendarização do ciclo de vida do projeto **(26/02/2020)**

**M2** - Apresentação da arquitetura do ciclo de vida do projeto: apresentação de uma arquitetura validada do sistema, descrevendo o funcionamento e fluxo de cada módulo do sistema a desenvolver **(11/03/2020)**

**M3** - Apresentação do protótipo intermédio: apresentação de um protótipo intermédio funcional (cerca de 80% das funcionalidades implementadas) do sistema **(15/04/2020)**

**M4** - Apresentação do projeto: apresentação do sistema final. Neste ponto, o sistema deve encontrar-se completamente funcional, isto é, com 100% das funcionalidades implementadas **(20/05/2020)**

## **Entregáveis**

Outputs relevantes do projeto são:

- I. Sistema funcional
  - A. API que disponibiliza os dados obtidos de modo a poderem ser utilizados noutros projetos;
  - B. Dashboard para visualização de consumos energéticos nos vários pisos do DETI;
  - C. Aplicação móvel com informação de ocupação de salas;
  - D. Software de integração dos sensores;
- II. Documentação dos diversos módulos (API, Sensores e Frontend);
- III. Microsite com informação da equipa e do projeto;

Todos os entregáveis, com identificador:

- E1. Apresentação do MVP do sistema (PowerPoint) - **M1**
- E2. Diagrama de Gantt do projeto (Diagrama) - **M1**
- E3. Microsite do projeto (Site) - **M1**
- E4. Documento de descrição de tarefas e entregáveis do projeto (PDF) - **M1**
- E5. Apresentação da arquitetura do sistema (PowerPoint) - **M2**
- E6. Documento de especificação da arquitetura do sistema (PDF) - **M2**
- E7. Apresentação do protótipo intermédio do sistema (PowerPoint) - **M3**
- E8. Protótipo intermédio do sistema (Produto funcional) - **M3**
- E9. Apresentação final do projeto (PowerPoint) - **M4**
- E10. Relatório técnico do sistema desenvolvido (PDF) - **M4**
- E11. Documentação online do funcionamento e utilização da API (Site) - **M4**