



INSTITUTO POLITÉCNICO DE LEIRIA
Projeto de Sistemas e Serviços

TeSP CRI | 2º ano

ENUNCIADO DO PROJETO

- VERSÃO AVALIAÇÃO PERIÓDICA -

OBJETIVOS

Objetivo deste projeto consiste em que os estudantes apliquem os conhecimentos adquiridos nas aulas teóricas e laboratoriais, de modo a definir e desenvolver um protótipo de uma solução de *Internet of Things* (IoT).

O projeto será implementado tendo por base as fichas laboratoriais da Unidade Curricular. De modo a otimizar o tempo das aulas laboratoriais os estudantes deverão planear e preparar as tarefas a realizar antes da última aula prática.

O protótipo a desenvolver deverá utilizar um cenário real ou virtual, através da utilização do software Cisco Packet Tracer (CPT), com a utilização de sensores, atuadores e controladores com capacidade de comunicação, de modo a interagir em tempo real com um servidor web, disponibilizando assim a informação *Anytime&Anywhere*.

As funcionalidades a implementar são as seguintes:

- Capacidade de atuação;
- Capacidade de *sensing*;
- Capacidade de transmissão e disponibilização da informação em tempo real;
- Disponibilização da informação *AnyTime&AnyWhere*;
- Capacidade de definição e envio de eventos por software (ou hardware);

Deverão ser apresentados todos os esquemas da solução IoT (Desenho Packet Tracer), o cenário de teste (como simularam por software um sistema real) e o protótipo a funcionar (no Packet Tracer, UniServerZ, Browser e Ambiente Windows em Python).

REQUISITOS MÍNIMOS DO PROJETO:

A proposta de projeto deve ser discutida com o docente da aula laboratorial e deve contemplar os seguintes requisitos:

EQUIPAMENTO/SOFTWARE OBRIGATORIOS	FUNCIONALIDADES OBRIGATÓRIAS
1x ou mais: <i>Single Board Computer (SBC)</i> no CPT (Cisco Packet Tracer)	<ul style="list-style-type: none"> - Deve estar conectado a uma rede local (cabo ou Wi-Fi); - Deve interligar sensores e/ou atuadores; - Código em <i>Python</i> CPT para controlo de sensores, atuadores, para enviar e receber comunicações HTTP;
1x ou mais: Microcontrolador (MCU) no CPT	<ul style="list-style-type: none"> - Deve estar conectado a uma rede local (cabo ou Wi-Fi); - Deve interligar sensores e/ou atuadores; - Código em <i>Python</i> CPT para controlo de sensores, atuadores, para enviar e receber comunicações HTTP;
1x ou mais: Switch no CPT	<ul style="list-style-type: none"> - Interligar equipamentos via cabo;
1x ou mais: Access Point (ou similar) no CPT	<ul style="list-style-type: none"> - Interligar equipamentos via Wi-Fi; - As configurações da rede devem ser seguras;
6x ou mais: Sensores no CPT	<ul style="list-style-type: none"> - Recolher dados do ambiente (<i>Environment</i>); - Dados recolhidos devem ser SEMPRE enviados para o serviço <i>web</i>, disponibilizando o valor atual na página Web;
6x ou mais: Atuadores no CPT	<ul style="list-style-type: none"> - Fazer atuar os atuadores; - Os estados dos atuadores devem ser SEMPRE enviados para o serviço <i>web</i>, disponibilizando o estado atual na página Web;
1x ou mais webcam real	<ul style="list-style-type: none"> - Deve ser utilizada para tirar fotografias e enviá-las para o serviço <i>web</i>, disponibilizando a última fotografia na página Web; - Pode ser utilizada para outras funcionalidades opcionais;
1 Website em HTML e PHP	<ul style="list-style-type: none"> - Desenvolver um website em HTML e PHP para mostrar o estado dos 6 sensores e dos 6 atuadores e que inclua: <ul style="list-style-type: none"> * API para recolha e envio de informações; * Página de histórico para cada sensores/atuadores; - Código HTML e PHP deve incluir o comprovativo de verificação de erros no código de acordo com a especificação de ambos os <i>standards</i>;

OUTROS REQUISITOS IMPORTANTES DO PROJETO:

Utilização de **múltiplos eventos** que despoletem ações utilizando a **comunicação via API**:

ORIGEM DOS EVENTOS	RESULTADO DOS EVENTOS
Eventos no código <i>Python</i>	Ações nos atuadores do CPT;
Eventos no código <i>Python</i>	Alterações dos conteúdos na página web;
Eventos no CPT	Alterações dos conteúdos na página web;
Eventos no CPT	Ações através do código <i>Python</i> (ex: <i>webcam</i> e/ou áudio);

Este projeto também envolve uma parte de pesquisa de soluções web que não foram desenvolvidas nas aulas práticas. Ou seja, os estudantes deverão pesquisar e implementar as seguintes funcionalidades em PHP e HTML:

1) Altere a API desenvolvida nas aulas de modo a que:

- a) seja realizada a autenticação de cada pedido HTTP, com uma **senha/token** de autenticação cujo valor é igual ao resultado do algoritmo de hashing MD5 da frase "**ProjetoCRI**".
- b) Todas as senhas/token inválidas devem devolver um HTTP *status_code* com um número de erro apropriado.
- c) Implemente um registo de *log* na API, de forma a guardar num ficheiro "*http_errors.log*" os pedidos HTTP que devolveram um HTTP *status_code* diferente de 200. Tenha em atenção as indicações seguintes:
 - Cada linha do ficheiro "*http_errors.log*", representa um pedido HTTP *status_code* com erro, conforme a sintaxe abaixo:

tipo_metodo_HTTP ; dados_no_pedido ; data_hora

Exemplo de get: GET;temperatura;2022/07/21 14:30

Exemplo de post: POST;temperatura;35.5;2022/07/21 14:30
 - Sugere-se que se utilize a estrutura de ficheiros idêntica à usada no projeto (crie uma diretoria *http_logs* para armazenar o ficheiro de log).

NOTA1: calcule o *hash* aqui: <https://md5calc.com/hash>

NOTA2: não confundir com a autenticação na *dashboard* (*login.php*)

Exemplo de um pedido HTTP com método GET: ***get(nome, token);***

Exemplo de um pedido HTTP com método POST: ***post(nome, datahora, valor, token);***

2) A página web inicial (index.php) deverá pedir um login e password que irá permitir entrar na página de visualização dos 6 sensores, 6 atuadores e do link ao historial.**3) O vosso relatório deverá ter uma parte de pesquisa (PS) sobre o método Let's Encrypt que permite instalar num determinado domínio um certificado digital gratuito, com validade de 3 meses. Por exemplo, se tivessem comprado o domínio www.aminhapagina_iot.pt, para colocar em funcionamento o vosso servidor web, como poderiam abrir a vossa página web com o protocolo https?**

IDEIAS DE PROJETO:

Cada grupo deverá escolher apenas 1 das seguintes ideias de projeto apresentadas na Tabela 1.

Cada grupo deve ser constituído por 3 estudantes.

#NOTA: Não pode haver 2 grupos de projeto a escolherem a mesma ideia de projeto.

Tabela 1 – Ideias de projeto para escolha

Nº	NOME DA IDEIA DE PROJETO
1	Lavandaria Inteligente
2	Padaria Inteligente
3	Apicultura Inteligente
4	Agência Inteligente de aluguer de autocarros
5	Clube aeronáutico Inteligente
6	Clube náutico Inteligente
7	Armazém robotizado Inteligente
8	Centro aduaneiro Inteligente
9	Canil Inteligente
10	Parque aquático Inteligente
11	Estação de ski Inteligente
12	Escola de condução Inteligente
13	Centro de testes COVID-19 inteligente

ENTREGA DO PROJETO

De acordo com o calendário de avaliação o Projeto deverá ser **entregue até à hora da data especificada, definida no calendário de avaliação.**

No **calendário de avaliação** está também a data e hora na qual é efetuada a **defesa oral do trabalho.**

Devem entregar na plataforma EAD a seguinte documentação (**5 elementos**):

- **Ficheiro Cisco Packet Tracer. O código** desenvolvido deve estar devidamente comentado;
- **Código da página web em HTML, Python e PHP**, devidamente comentado;
- **Código Python**, devidamente comentado;

- **Vídeo simples** com a demonstração das potencialidades do protótipo (no máximo 100 MB); Devem utilizar o URL seguinte para comprimir ficheiros de vídeo: <http://www.videosmaller.com/>
- **Relatório** escrito onde deve estar toda a informação relevante do Projeto, as decisões mais importantes, esquemas da solução, esquemas de rede e esquema de blocos da arquitetura IoT. Deve ser utilizado o **template** disponível no portal ead.ipleiria.pt

O relatório deve ser disponibilizado apenas em **formato digital PDF** e ter a seguinte estrutura:

1. Objetivos, com o possível enquadramento da solução numa situação real concreta;
2. Descrição e arquitetura da solução e os equipamentos utilizados;
3. Implementação (aspetos mais importantes);
4. Cenário de teste;
5. Resultados obtidos;
6. Pesquisa sobre Let's Encrypt para ativar o protocolo HTTPS;
7. Conclusão (análise da solução, potencialidades e pontos a melhorar);
8. Bibliografia;
9. Anexos com o código (um Anexo por equipamento/módulo: MCU, SBC e Web).

CLASSIFICAÇÃO FINAL

A classificação final (**CF**) será determinada de acordo com a seguinte fórmula:

$$CF = (40\%WORK + 20\%WEB + 10\%SEG + 20\%REL + 5\%FX + 5\%PS) \times DO$$

As classificações parcelares (0-20 valores) correspondem a:

- **WORK** – Protótipo com os requisitos mínimos incluindo os conceitos de *python*;
- **WEB** – Página Web com a aplicação dos conceitos de HTML, CSS e PHP;
- **SEG** – Implementação do Token na API e implementação do Login com password;
- **REL** – Relatório escrito;
- **FX** – Funcionalidades extras implementadas e que não foram requeridas neste enunciado;
- **PS** – Pesquisa sobre segurança https com Let's Encrypt;
- **DO** – Defesa **oral** individual (0% ou 100%);

A entrega de documentação (relatório, código, vídeo da aplicação) e a defesa oral são obrigatórias.