

## Arquitetura de Computadores 2025/2026 – MEEC – ISEP

### Mini-projeto

O mini-projeto deverá ser realizados preferencialmente em **grupo de dois estudantes**. Não serão permitidos mais do que dois elementos por grupo. A eventual realização de trabalhos individuais deverá ser discutida com o docente das aulas PL.

O mini-projeto é realizado fora do período das aulas.

### Datas Importantes

- **O mais brevemente possível:** cada grupo deverá enviar um *email* para [jes@isep.ipp.pt](mailto:jes@isep.ipp.pt) indicando os nomes dos elementos do grupo e a lista de trabalhos preferenciais. **As preferências serão atendidas por ordem de chegada.**
- **23 de dezembro de 2025** – data-limite para entrega do **relatório final**. O relatório deverá ser enviado em **formato PDF** para [jes@isep.ipp.pt](mailto:jes@isep.ipp.pt). **Nessa altura, deverá ser partilhada, via conta OneDrive do ISEP de um dos elementos do grupo, uma imagem do conteúdo do dispositivo de armazenamento do sistema final comprimida em formato xz.**

### Lista de Projetos Propostos

Salienta-se que não é disponibilizado qualquer equipamento, pelo que os estudantes deverão escolher as propostas que se adequam mais aos meios de que dispõem.

	Projeto	Comentários
1	Moldura eletrónica para Raspberry Pi	Apresentação sequencial em ecrã inteiro de imagens em formato PNG. As imagens deverão ser obtidas por HTTP (por exemplo, <code>wget -N</code> ).
2	Sistema de vigilância	Utilização de câmara USB, armazenamento de imagens, <i>streaming</i> . Identificar no relatório qual ou quais os módulos de <i>kernel</i> usados para a(s) câmara(s).
3	Servidor de ficheiros	Deverá oferecer acesso via Samba, SFTP e consola por SSH. Como valorização, utilizar uma configuração RAID (esta componente pode ser simulada em ambiente virtual).
4	<i>Audio Player/ Media center</i>	Exemplos: mplayer, Kodi, rádio Internet (identificar no relatório quais os módulos de <i>kernel</i> usados para o áudio e/ou vídeo)
5	<i>Data logger</i>	Análise de soluções de base de dados para sistemas dedicados. Implementação de solução. Simular a aquisição de dados através de interfaces do computador.
6	Web browser para Raspberry Pi	Criação de uma distribuição mínima com <i>web browser</i> para Raspberry Pi modelo 3 (ou mais recente) e respetivo teste em QEMU. O <i>web browser</i> deverá preferencialmente ser iniciado automaticamente, sem login de utilizador, e ocupar o ecrã inteiro.

7	Yocto Project	Criação de distribuição Linux usando as ferramentas do Yocto Project ( <a href="https://www.yoctoproject.org/">https://www.yoctoproject.org/</a> ). Criar sistema básico apenas com shell. Ilustrar o procedimento de adição de serviços (ssh, http, por exemplo).
8	LXC/LXD	Criação de <i>containers</i> LXC para Fedora sem recorrer a <i>templates</i> pré-definidos. O <i>container</i> final deverá demonstrar a execução do servidor HTTP. Confrontar com solução baseada em Podman/Docker.
9	Análise de módulo de sistema de ficheiros	O sistema de ficheiros pode ser um dos suportados oficialmente pelo Linux (ext2, por exemplo) ou um desenvolvido por terceiros (simpleFS, por exemplo)
10	Análise do código fonte de controlador de rede	virtio-net ou qualquer controlador <i>ethernet</i> suportado pelo Linux
11	Análise do subsistema de GPIO do Linux no Raspberry Pi	Identificação dos módulos utilizados no Raspberry Pi modelo 3 (ou superior) e análise do respetivo código fonte.
12	Análise das potencialidades “real-time” do Linux 6.12 (PREEMPT_RT)	Criar distribuição baseado nessa versão do kernel e obter estáticas de latências relativamente a comutação de tarefas (por exemplo, com <a href="https://github.com/akopytov/sysbench">https://github.com/akopytov/sysbench</a> ou <a href="https://github.com/gchamp20/RTOSBench">https://github.com/gchamp20/RTOSBench</a> )

Poderão também ser aceites propostas de trabalho por parte dos estudantes. Para tal, os elementos do grupo deverão enviar uma proposta resumida do tópico que desejam abordar.

### Instruções adicionais

Nos projetos que consistem na criação de um sistema baseado em Linux dedicado a uma determinada aplicação (propostas 1 a 7), o sistema final deverá conter apenas os ficheiros necessários para o funcionamento da mesma. Sempre que possível, o *software* deverá ser compilado a partir do código fonte, podendo utilizar a ferramenta Buildroot, se aplicável.

Podem também ser realizados testes preliminares do *software* numa distribuição Linux standard (Fedora, Ubuntu, etc.). Esta abordagem pode ser útil para despistar eventuais problemas de configuração.

Fatores de valorização do mini-projeto, que deverão ser descritas no relatório, incluem: configuração do *kernel* à medida da aplicação, configuração das aplicações, configuração de *scripts* de arranque do sistema, desenvolvimento de programas, acessibilidade/usabilidade através da rede de dados e utilização de *hardware*.

O relatório do trabalho deverá obedecer às seguintes diretivas:

- Deverá ser o mais sucinto possível, não esquecendo, contudo, as regras elementares de edição (especialmente o correto uso da ortografia e gramática).
- Deverá apresentar um guia que permita reproduzir todos os passos necessários para a preparação e execução do trabalho (*download* de ficheiros, instalações de pacotes, sistemas de desenvolvimento, *scripts* de arranque, configurações diversas, etc.).
- Deverá apresentar as estatísticas de utilização de memória RAM, assim como utilização de espaço no dispositivo de armazenamento. O espaço ocupado deverá ser categorizado por diretórios e sucintamente justificado.
- Deverá apresentar impressões do ecrã representativas dos ensaios realizados.
- Todas as fontes documentais usadas para a elaboração do trabalho deverão ser citadas no relatório (inclusive programas exemplificativos e tutoriais).