

**PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO - PROPESP**  
**COORDENAÇÃO DE PESQUISA E PROJETOS INSTITUCIONAIS - CPPI**  
**PROGRAMA DE INICIAÇÃO CIENTÍFICA E TECNOLÓGICA DA UNIVERSIDADE DO**  
**ESTADO DO AMAZONAS**

RELATÓRIO MENSAL			
<b>EDITAL ICT</b>	( X ) Edital Nº 031/2025 - GR/UEA		
<b>TIPO DE PROJETO</b>	<input type="checkbox"/> PBICT-Af/UEA <input checked="" type="checkbox"/> PAIC/FAPEAM <input type="checkbox"/> PIBIC/CNPq <input type="checkbox"/> PIBITI/CNPq <input type="checkbox"/> PIBIC Af/CNPq <input type="checkbox"/> VOLUNTÁRIO	<b>SISPROJ:</b>	59617
<b>TÍTULO DO PROJETO:</b>	Sistema de Detecção de Quedas Utilizando Inteligência Artificial e Alerta Automatizado		
<b>ALUNO:</b>	Nelson Emeliano Silva		
<b>ORIENTADOR:</b>	Angilberto Muniz Ferreira Sobrinho		
<b>MÊS DE REFERÊNCIA DO RELATÓRIO</b>	Setembro/2025	<b>DATA DE ENTREGA DO RELATÓRIO</b>	29/09/2025

**ATIVIDADES DESENVOLVIDAS**

**1. Resumo das Atividades do Período**

Durante o segundo mês, foi finalizada a fase de fundamentação teórica e coleta de referências e iniciada a fase de implementação prática. O principal foco do período foi a estruturação completa do ambiente de desenvolvimento e a consolidação da base de dados para o treinamento do modelo. Foi estabelecido um ambiente de programação personalizado e isolado, e foram coletados e pré-analizados os conjuntos de dados que servirão como base para o desenvolvimento do sistema de detecção.

**2. Detalhamento das Atividades Realizadas**

**2.1. Conclusão da Revisão Bibliográfica**

A pesquisa da literatura técnica foi finalizada neste período. O estudo resultou em um referencial teórico que valida a abordagem metodológica proposta, especialmente o uso de Redes Neurais Convolucionais (CNNs) para a extração de características visuais e de Redes Neurais Recorrentes (LSTMs) para a análise da sequência temporal dos movimentos.

**2.2. Estruturação do Ambiente de Desenvolvimento**

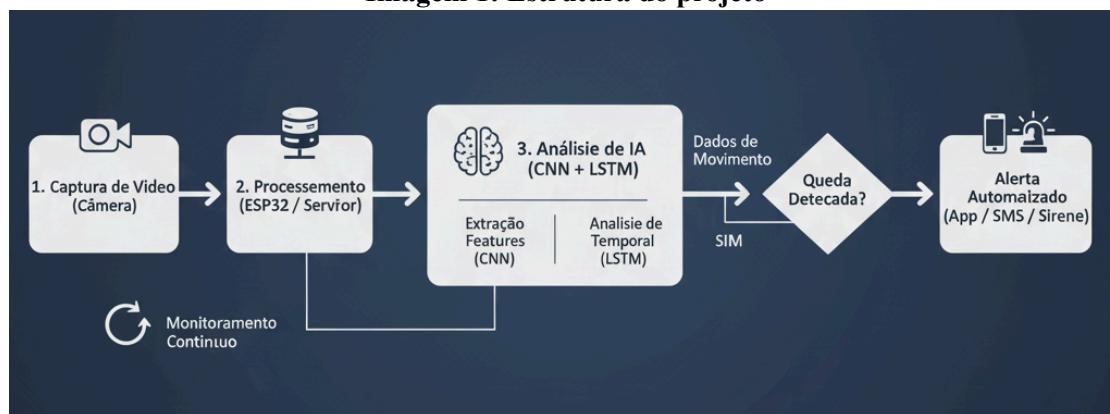
Foi configurado um ambiente de desenvolvimento padronizado sobre o sistema operacional Linux (Ubuntu 24 LTS). Para garantir o gerenciamento controlado das dependências e evitar conflitos, foi estabelecido um ambiente virtual Python (venv) dentro da estrutura de diretórios do projeto.

Com o ambiente isolado e ativado, procedeu-se à instalação das bibliotecas de software essenciais. Foram instaladas as bibliotecas TensorFlow e PyTorch para o desenvolvimento dos modelos de aprendizado de máquina, e a biblioteca OpenCV para as tarefas de processamento de imagem e visão computacional. O Visual Studio Code foi configurado como o editor de código principal, com o interpretador Python apontando para o ambiente virtual do projeto, assegurando a correta utilização das bibliotecas instaladas. Por fim, o projeto foi inicializado como um repositório Git para o controle de versão do código-fonte.

### 2.3. Definindo o Escopo e Estrutura do Projeto

Este diagrama descreve as etapas principais do projeto, desde a captura da imagem até o envio do alerta automatizado.

**Imagen 1: Estrutura do projeto**



O fluxo, conforme ilustrado, segue as seguintes etapas:

- Captura de Vídeo:** Um monitoramento contínuo é realizado por uma câmera.
- Processamento:** O fluxo de vídeo é processado, seja por um microcontrolador (ESP32) ou um servidor, para preparar os dados para a análise.
- Análise de IA:** O modelo de CNN extrai as características visuais de cada quadro e o modelo LSTM analisa a sequência temporal para identificar padrões de movimento consistentes com uma queda.
- Decisão e Alerta:** Se uma queda é detectada, um alerta automatizado é disparado para um aplicativo, SMS ou alarme sonoro.

### 2.4. Coleta e Preparação dos Datasets

A etapa de coleta de dados envolveu a obtenção de múltiplos conjuntos de dados para garantir a diversidade e a robustez do treinamento. Foram adquiridos os seguintes datasets:

- O **UR Fall Detection Dataset**, obtido de seu repositório oficial na Universidade de Rzeszów, servirá como base principal de vídeos contendo quedas simuladas e atividades cotidianas. ([UR Fall Detection Dataset](#))

- Uma versão do mesmo dataset está sendo explorada na plataforma **Roboflow**, que oferece ferramentas para agilizar o pré-processamento e a aplicação de técnicas de *data augmentation*. ([UR Fall Detection Dataset Model > Overview](#))
- Como fonte de dados complementar, foi obtido o **Fall Detection Dataset** da plataforma Kaggle, que consiste em um conjunto de imagens estáticas já classificadas, útil para aumentar o volume de dados de treinamento. ([Fall Detection Dataset](#))

Foi iniciado o desenvolvimento de um script para o pré-processamento, cuja função é extrair os quadros dos vídeos, normalizar as dimensões das imagens e organizá-las em uma estrutura de diretórios adequada para o treinamento do modelo.

### Imagen 2 - Trecho do código inicial para teste do modelo CNN-LTSM

```
C:\> Users > Nelsinho > OneDrive > Área de Trabalho > Projetos > UEA > PAIC > inicial.py > build_cnn_lstm_model

1 # Importação das bibliotecas necessárias
2 import tensorflow as tf
3 from tensorflow import keras
4 from tensorflow.keras import layers
5 import numpy as np
6
7 SEQUENCE_LENGTH = 20
8 IMG_HEIGHT = 224
9 IMG_WIDTH = 224
10 CHANNELS = 3
11
12 def build_cnn_lstm_model():
13     """
14         Constrói o modelo híbrido CNN-LSTM para detecção de quedas.
15     """
16     print("Construindo o modelo...")
17
18     # --- 2. Definição da Arquitetura ---
19     input_shape = (SEQUENCE_LENGTH, IMG_HEIGHT, IMG_WIDTH, CHANNELS)
20     inputs = keras.Input(shape=input_shape)
21
22     # --- Extrator de Características Visuais ---
23     base_model = keras.applications.MobileNetV2(
24         include_top=False,
25         weights='imagenet',
26         input_shape=(IMG_HEIGHT, IMG_WIDTH, CHANNELS)
27     )
28
29     base_model.trainable = False
30     cnn_output = layers.TimeDistributed(base_model)(inputs)
31
32     # --- Analisador de Sequência Temporal ---
33     lstm_input = layers.TimeDistributed(layers.GlobalAveragePooling2D())(cnn_output)
34
```

### 3. Resultados do Período

- **Referencial Teórico Consolidado:** Literatura consolidada, que servirá de suporte para as próximas etapas do projeto.
- **Ambiente de Desenvolvimento Operacional:** Um ambiente de programação configurado, pronto para o desenvolvimento dos algoritmos.
- **Base de Dados Consolidada:** Um conjunto de dados diversificado, proveniente de múltiplas fontes, foi coletado e está em fase inicial de preparação.
- **Repositório de Código Versionado:** O projeto está sob controle de versão com Git, garantindo a rastreabilidade do desenvolvimento.  
(<https://github.com/Nelson-esilva/Fall-Detect-System.git>)

## AVALIAÇÃO DO ORIENTADOR

Deverá ser preenchido um relatório mensal pelo (a) aluno (a) e orientador (a), informando as atividades desenvolvidas e anexado no SISPROJ até o dia 30 de cada mês, a partir do mês de implementação do projeto. O acompanhamento do relatório e frequência mensais será realizado pelo Comitê Local.



**UEA**  
UNIVERSIDADE  
DO ESTADO DO  
AMAZONAS