

# App: Calculadora de IMC

Por Lorena F. Izidoro e Luis Felipe G. de S. Paim

# Dificuldades encontradas

- Esse projeto apresentou uma complexidade bem maior que o anterior, exigindo o uso de diversos arquivos e uma estruturação intrincada.
- Principal questão: implementação da exportação de dados via arquivo .csv

```
...
val writer = FileWriter(file)

writer.append("Data,Peso,Altura,IMC,Classificacao,T
MB,TDEE\n")

history.forEach { item ->
    writer.append("${item.getFormattedDate()}")
    writer.append("${item.weight},")
    writer.append("${item.height},")
    writer.append("${String.format("%.2f",
item.imc)},")
    writer.append("${item.classification},")
    writer.append("${item.tmb},")
    writer.append("${item.tdee}\n")
}
writer.flush()
writer.close()
...
```

# LLMs utilizadas

- Nome do modelo: Gemini
- Exemplo de prompt: “Aja como um especialista em kotlin com muitos anos de experiência na área. Com base nisso, como você implementaria uma solução para um aplicativo de cálculo de IMC que usa compose e arquitetura MVC para exportar em um csv o histórico das medições?”
- A LLM auxiliou muito para que o trabalho fosse concluído dentro do prazo, principalmente no desenvolvimento Front End e na função de exportação de dados.

```
...
Button(
    onClick = {
        if (historyListForExport.isNotEmpty())
            CsvExporter.exportAndShare(context,
                historyListForExport)
        else Toast.makeText(context, "Histórico vazio!",
            Toast.LENGTH_SHORT).show()
    },
    modifier = Modifier
        .weight(1f)
        .height(50.dp),
    colors = ButtonDefaults.buttonColors(
        containerColor = Color.White, // Botão Branco
        contentColor = HealthSecondary // Texto Verde
    )
...
}
```