

Projeto: Plano de Aula — IA na Educação Física Escolar

Autor: Matheus Florindo · Claude AI (Anthropic)

Data: 2026-02-25

Modelo IA: Claude Opus 4.6

Plataforma: Cowork Mode — Claude Desktop

Linguagem: Python 3.10

Biblioteca PDF: ReportLab 4.4.10

Total linhas: 1920

Total arquivos: 2

Dependências: reportlab==4.4.10, pypdf

Cmd instalação: pip install reportlab --break-system-packages

Cmd execução 1: python3 plano_aula_premium.py

Cmd execução 2: python3 doc_tecnica_premium.py

Script principal — Geração do PDF Ultra Premium do Plano de Aula com IA na Educação Física Escolar

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 PDF Ultra Premium – Plano de Aula Institucional
5 Design de alto nível: cabeçalho/rodapé com faixa, tipografia profissional.
6 """
7
8 from reportlab.lib.pagesizes import A4
9 from reportlab.lib import colors
10 from reportlab.lib.units import mm, cm
11 from reportlab.lib.styles import ParagraphStyle
12 from reportlab.lib.enums import TA_LEFT, TA_CENTER, TA_RIGHT, TA_JUSTIFY
13 from reportlab.platypus import (
14     SimpleDocTemplate, Paragraph, Spacer, Table, TableStyle,
15     HRFlowable, KeepTogether, PageBreak
16 )
17 from reportlab.pdfgen import canvas
18 from reportlab.platypus.flowables import Flowable
19 from reportlab.lib.colors import HexColor
20 import os
21
22 # ■■■ PALETA DE CORES ■■■
23 NAVY = HexColor("#0D1B2A") # Azul escuro institucional
24 NAVY_MED = HexColor("#1B3A5C") # Azul médio
25 TEAL = HexColor("#0E7C7B") # Verde-azulado accent
26 GOLD = HexColor("#C9A227") # Dourado accent
27 GOLD_LIGHT = HexColor("#F0D080") # Dourado claro
28 WHITE = HexColor("#FFFFFF")
29 LIGHT_BG = HexColor("#F4F7FB") # Fundo seções
30 LIGHT_BG2 = HexColor("#EAF0F8")
31 BORDER = HexColor("#BDD0E8")
32 TEXT_DARK = HexColor("#0D1B2A")
33 TEXT_MED = HexColor("#2C4E6C")
34 TEXT_LIGHT = HexColor("#5A7A99")
35 ACCENT_RED = HexColor("#C0392B")
36 GREEN_OK = HexColor("#1A7A4A")
37 STRIPE_1 = HexColor("#E8F0F8")
38 STRIPE_2 = HexColor("#FFFFFF")
39
40 W, H = A4 # 595.28 x 841.89 pts
41 MARGIN_L = 2.0 * cm
42 MARGIN_R = 2.0 * cm
43 MARGIN_T = 3.2 * cm # Espaço para header
44 MARGIN_B = 2.8 * cm # Espaço para footer
45
46 OUTPUT = "/sessions/wonderful-bold-ptolemy/mnt/outputs/Plano_Aula_IA_EF_Premium.pdf"
47
48 # ■■■ HEADER & FOOTER CANVAS ■■■
49 def draw_header_footer(canv, doc):
50     canv.saveState()
51     page_num = doc.page
52
53     # ■■ HEADER ■■■
54     # Faixa superior total (navy escuro)
55     canv.setFillColor(NAVY)
56     canv.rect(0, H - 22*mm, W, 22*mm, fill=1, stroke=0)
57
58     # Faixa accent dourada (linha fina embaixo do navy)
59     canv.setFillColor(GOLD)
60     canv.rect(0, H - 24*mm, W, 2*mm, fill=1, stroke=0)

```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

619 for bg, titulo, itens in conceitos:
620     header_row = [[Paragraph(titulo, ParagraphStyle("ch", fontName="Helvetica-Bold",
621         fontSize=8.5, textColor=WHITE))]]
622     rows = [[Paragraph(f"• {item}", ParagraphStyle("ci", fontName="Helvetica",
623         fontSize=8, leading=12, textColor=TEXT_MED))]] for item in itens]
624     all_rows = header_row + rows
625     t = Table(all_rows, colWidths=[col_w])
626     style_list = [
627         ("BACKGROUND", (0,0), (-1,0), bg),
628         ("BACKGROUND", (0,1), (-1,-1), LIGHT_BG),
629         ("LEFTPADDING", (0,0), (-1,-1), 10),
630         ("RIGHTPADDING", (0,0), (-1,-1), 10),
631         ("TOPPADDING", (0,0), (-1,-1), 5),
632         ("BOTTOMPADDING", (0,0), (-1,-1), 5),
633         ("LINEBELOW", (0,0), (-1,-1), 0.3, BORDER),
634         ("BOX", (0,0), (-1,-1), 0.5, bg),
635     ]
636     t.setStyle(TableStyle(style_list))
637     story.append(t)
638     story.append(Spacer(1, 3))
639
640 story.append(Spacer(1, 5*mm))
641
642 # Sub 4.3 Atividade Prática – Tabela de Dados
643 story.append(Paragraph("4.3 Atividade Prática Orientada – Tabela de Dados Fictícios", styles["section_title_sm"]))
644 story.append(Paragraph(
645     "Os estudantes analisam a tabela abaixo e respondem às perguntas guiadas. "
646     "Os dados são fictícios e utilizados exclusivamente para fins pedagógicos, "
647     "em conformidade com os princípios da LGPD.",
648     styles["body"]))
649 story.append(Spacer(1, 4))
650
651 tbl_headers = ["ALUNO", "TEMPO", "FC MÉDIA", "PASSOS", "ANÁLISE"]
652 tbl_data_rows = [
653     ["A", "12 min", "165 bpm", "1.800", "Alta eficiência metabólica"],
654     ["B", "10 min", "178 bpm", "2.100", "Alto esforço cardiovascular"],
655     ["C", "11 min", "155 bpm", "1.700", "Melhor equilíbrio esforço/tempo"],
656 ]
657 header_p = [Paragraph(h, styles["tbl_header"]) for h in tbl_headers]
658 data_table = [header_p]
659 for row in tbl_data_rows:
660     data_table.append([Paragraph(c, styles["tbl_cell"]) for c in row])
661
662 aluno_col = 1.5*cm
663 other_cols = [(col_w - aluno_col) / (len(tbl_headers)-1)] * (len(tbl_headers)-1)
664 dt = Table(data_table, colWidths=[aluno_col] + other_cols)
665 dt.setStyle(TableStyle([
666     ("BACKGROUND", (0,0), (-1,0), NAVY),
667     ("ROWBACKGROUNDS", (0,1), (-1,-1), [STRIPE_1, STRIPE_2, HexColor("#F0F8F0")]),
668     ("ALIGN", (0,0), (-1,-1), "CENTER"),
669     ("VALIGN", (0,0), (-1,-1), "MIDDLE"),
670     ("TOPPADDING", (0,0), (-1,-1), 7),
671     ("BOTTOMPADDING", (0,0), (-1,-1), 7),
672     ("LINEBELOW", (0,0), (-1,-1), 0.4, BORDER),
673     ("LINEBEFORE", (0,0), (0,-1), 3, TEAL),
674     ("BOX", (0,0), (-1,-1), 0.8, NAVY),
675 ]))
676 story.append(dt)
677 story.append(Spacer(1, 4))
678 story.append(Paragraph(
679     "Tabela 1 – Dados fictícios para análise pedagógica. Fonte: elaboração própria para fins educacionais.",
680     styles["caption"]))
681 story.append(Spacer(1, 4))
682
683 # Perguntas guiadas
684 perguntas = [
685     ("P1", "Quem teve maior esforço cardiovascular? Justifique com base nos dados."),
686     ("P2", "Quem apresentou melhor eficiência (menor tempo e menor FC)? Por quê?"),
687     ("P3", "Qual variável você considera mais relevante para avaliar desempenho? Argumente."),
688 ]
689 pq_rows = []
690 for cod, texto in perguntas:
691     pq_rows.append([
692         Paragraph(cod, ParagraphStyle("pc", fontName="Helvetica-Bold",
693             fontSize=8.5, textColor=WHITE, alignment=TA_CENTER)),
694         Paragraph(texto, ParagraphStyle("pt", fontName="Helvetica",
695             fontSize=8.5, leading=12, textColor=TEXT_DARK)),
696     ])
697 pq_t = Table(pq_rows, colWidths=[1.2*cm, col_w - 1.2*cm])
698 pq_t.setStyle(TableStyle([
699     ("BACKGROUND", (0,0), (-1,-1), LIGHT_BG),
700     ("BACKGROUND", (0,0), (0,0), GOLD),
701     ("BACKGROUND", (0,1), (0,1), TEAL),
702     ("BACKGROUND", (0,2), (0,2), NAVY_MED),
703     ("VALIGN", (0,0), (-1,-1), "MIDDLE"),
704     ("LEFTPADDING", (0,0), (-1,-1), 8),
705     ("RIGHTPADDING", (0,0), (-1,-1), 8),
706     ("TOPPADDING", (0,0), (-1,-1), 7),
707     ("BOTTOMPADDING", (0,0), (-1,-1), 7),
708     ("LINEBELOW", (0,0), (-1,-1), 0.3, BORDER),
709 ]))
710 story.append(pq_t)
711 story.append(Spacer(1, 6*mm))

```



```

712
713 #
714 # SEÇÃO 5 – DISCUSSÃO ÉTICA (LGPD)
715 #
716 for e in section_header("05 · Dimensão Ética", "Proteção de Dados e LGPD", styles):
717     story.append(e)
718
719 etica_data = [[Paragraph(
720     "<b>Base Legal:</b> Lei Geral de Proteção de Dados Pessoais – Lei 13.709/2018",
721     ParagraphStyle("el", fontName="Helvetica-Bold", fontSize=8.5,
722         textColor=WHITE))]]
723 etica_header = Table(etica_data, colWidths=[col_w])
724 etica_header.setStyle(TableStyle([
725     ("BACKGROUND", (0,0), (-1,-1), HexColor("#7B3F00")),
726     ("LEFTPADDING", (0,0), (-1,-1), 12),
727     ("TOPPADDING", (0,0), (-1,-1), 8),
728     ("BOTTOMPADDING", (0,0), (-1,-1), 8),
729 ]))
730 story.append(etica_header)
731
732 reflexoes = [
733     ("Consentimento", "Os dados pessoais de estudantes, especialmente menores, exigem autorização expressa do ..."),
734     ("Armazenamento", "Dados biométricos e de desempenho devem ser armazenados com segurança, por período determi..."),
735     ("Finalidade", "O uso dos dados deve ser estritamente pedagógico, não podendo ser reaproveitado para fi..."),
736     ("Transparência", "Os estudantes e responsáveis devem ser informados sobre quais dados são coletados, como...")
737 ]
738 for titulo, desc in reflexoes:
739     row = [
740         Paragraph(titulo, ParagraphStyle("rt", fontName="Helvetica-Bold",
741             fontSize=8.2, textColor=HexColor("#7B3F00"))),
742         Paragraph(desc, ParagraphStyle("rd", fontName="Helvetica",
743             fontSize=8, leading=12, textColor=TEXT_MED)),
744     ]
745     t = Table(row, colWidths=[2.8*cm, col_w - 2.8*cm])
746     t.setStyle(TableStyle([
747         ("BACKGROUND", (0,0), (-1,-1), HexColor("#FDF3E7")),
748         ("LEFTPADDING", (0,0), (-1,-1), 10),
749         ("RIGHTPADDING", (0,0), (-1,-1), 10),
750         ("TOPPADDING", (0,0), (-1,-1), 6),
751         ("BOTTOMPADDING", (0,0), (-1,-1), 6),
752         ("LINEBELOW", (0,0), (-1,-1), 0.3, BORDER),
753         ("LINEAFTER", (0,0), (0,-1), 0.8, HexColor("#7B3F00")),
754     ]))
755     story.append(t)
756 story.append(Spacer(1, 6*mm))
757
758 #
759 # SEÇÃO 6 – DIFERENCIAÇÃO PEDAGÓGICA
760 #
761 for e in section_header("06 · Diferenciação", "Pedagógica Estruturada", styles):
762     story.append(e)
763
764 perfis = [
765     (TEAL, "Aluno com Dificuldade",
766         ["Uso de gráficos visuais e representações pictóricas",
767          "Explicação oral mediada pelo professor",
768          "Simplificação dos dados (2 variáveis)"]),
769     (NAVY_MED, "Perfil Verbal / Argumentativo",
770         ["Debate argumentativo sobre eficiência vs. esforço",
771          "Defesa oral de posição com base em dados",
772          "Redação de parágrafo justificativo"]),
773     (GOLD, "Aluno Avançado / Perfil Analítico",
774         ["Proposta de modelo simples de pontuação ponderada",
775          "Cálculo comparativo com pesos diferenciados",
776          "Extensão: introdução ao conceito de normalização"]),
777 ]
778 perf_rows = []
779 header_row = [
780     Paragraph(t, ParagraphStyle("ph", fontName="Helvetica-Bold",
781         fontSize=8, textColor=WHITE, alignment=TA_CENTER))
782     for _, t, _ in perfis
783 ]
784 perf_rows.append(header_row)
785
786 max_items = max(len(i) for _, _, i in perfis)
787 for j in range(max_items):
788     row = []
789     for _, _, items in perfis:
790         if j < len(items):
791             row.append(Paragraph(f"• {items[j]}", ParagraphStyle("pi",
792                 fontName="Helvetica", fontSize=8, leading=12, textColor=TEXT_MED)))
793         else:
794             row.append(Paragraph("", styles["body"]))
795     perf_rows.append(row)
796
797 perf_t = Table(perf_rows, colWidths=[col_w/3]*3)
798 ts_list = [
799     ("ROWBACKGROUNDS", (0,1), (-1,-1), [STRIPE_1, STRIPE_2]),
800     ("ALIGN", (0,0), (-1,0), "CENTER"),
801     ("VALIGN", (0,0), (-1,-1), "TOP"),
802     ("LEFTPADDING", (0,0), (-1,-1), 8),
803     ("RIGHTPADDING", (0,0), (-1,-1), 8),
804     ("TOPPADDING", (0,0), (-1,-1), 6),

```

[illegible]

[illegible]

[illegible]

Script de documentação técnica — Gera o PDF com Stack Tecnológica, Prompt Exato e Código-Fonte

[illegible]

[illegible]

```

142         fontSize=7.5, leading=10, textColor=GOLD_LIGHT, backColor=CODE_BG2)
143     s["meta"] = ParagraphStyle("m", fontName="Helvetica", fontSize=7.5,
144         leading=10, textColor=TEXT_LIGHT, alignment=TA_CENTER, spaceAfter=1)
145     return s
146
147
148 def divider(color=TEAL, thickness=0.8):
149     return HRFlowable(width="100%", thickness=thickness, color=color,
150         spaceBefore=4, spaceAfter=4)
151
152 def sec(label, title, styles):
153     return [Spacer(1, 5),
154         Paragraph(label.upper(), styles["section_label"]),
155         Paragraph(title, styles["section_title"]),
156         divider(TEAL, 1.2)]
157
158 def code_block(title, code_text, styles):
159     """Bloco de código estilizado."""
160     elements = []
161     # Title bar
162     title_data = [[Paragraph(f" {title}", styles["code_title"])]
163     tt = Table(title_data, colWidths=[COL_W])
164     tt.setStyle(TableStyle([
165         ("BACKGROUND", (0,0), (-1,-1), CODE_BG2),
166         ("TOPPADDING", (0,0), (-1,-1), 5),
167         ("BOTTOMPADDING", (0,0), (-1,-1), 5),
168         ("LEFTPADDING", (0,0), (-1,-1), 8),
169     ]))
170     elements.append(tt)
171
172     # Code content - split into lines
173     lines = code_text.strip().split("\n")
174     code_rows = []
175     for i, line in enumerate(lines):
176         safe_line = line.replace("&", "&amp;").replace("<", "&lt;").replace(">", "&gt;")
177         if not safe_line.strip():
178             safe_line = " "
179         bg = CODE_BG if i % 2 == 0 else HexColor("#1A2744")
180         num_style = ParagraphStyle("ln", fontName="Courier", fontSize=5.8,
181             leading=8, textColor=HexColor("#475569"), alignment=TA_RIGHT)
182         code_style = ParagraphStyle("cl", fontName="Courier", fontSize=6.2,
183             leading=8.5, textColor=HexColor("#E2E8F0"))
184         code_rows.append([
185             Paragraph(str(i+1), num_style),
186             Paragraph(safe_line, code_style),
187         ])
188
189     if code_rows:
190         ct = Table(code_rows, colWidths=[0.8*cm, COL_W - 0.8*cm])
191         ct_styles = [
192             ("LEFTPADDING", (0,0), (-1,-1), 4),
193             ("RIGHTPADDING", (0,0), (-1,-1), 4),
194             ("TOPPADDING", (0,0), (-1,-1), 1),
195             ("BOTTOMPADDING", (0,0), (-1,-1), 1),
196             ("LINEBEFORE", (0,0), (0,-1), 0.5, HexColor("#334155")),
197         ]
198         for i in range(len(code_rows)):
199             bg = CODE_BG if i % 2 == 0 else HexColor("#1A2744")
200             ct_styles.append(("BACKGROUND", (0,i), (-1,i), bg))
201         ct.setStyle(TableStyle(ct_styles))
202         elements.append(ct)
203
204     # Bottom border
205     elements.append(HRFlowable(width="100%", thickness=2, color=TEAL,
206         spaceBefore=0, spaceAfter=6))
207     return elements
208
209
210 def tech_row(sigla, bg, nome, descricao, versao, styles):
211     row = [
212         Paragraph(sigla, ParagraphStyle("s", fontName="Helvetica-Bold",
213             fontSize=9, textColor=WHITE, alignment=TA_CENTER)),
214         Paragraph(nome, ParagraphStyle("n", fontName="Helvetica-Bold",
215             fontSize=8.5, textColor=NAVY, spaceAfter=1)),
216         Paragraph(descricao, ParagraphStyle("d", fontName="Helvetica",
217             fontSize=8, leading=12, textColor=TEXT_MED, alignment=TA_JUSTIFY)),
218         Paragraph(versao, ParagraphStyle("v", fontName="Courier-Bold",
219             fontSize=8, textColor=TEAL, alignment=TA_CENTER)),
220     ]
221     t = Table(row, colWidths=[1.4*cm, COL_W - 3.6*cm, 2.2*cm])
222     t.setStyle(TableStyle([
223         ("BACKGROUND", (0,0), (0,0), bg),
224         ("BACKGROUND", (1,0), (-1,0), LIGHT_BG),
225         ("VALIGN", (0,0), (-1,-1), "MIDDLE"),
226         ("LEFTPADDING", (0,0), (-1,-1), 8),
227         ("RIGHTPADDING", (0,0), (-1,-1), 8),
228         ("TOPPADDING", (0,0), (-1,-1), 7),
229         ("BOTTOMPADDING", (0,0), (-1,-1), 7),
230         ("LINEBELOW", (0,0), (-1,-1), 0.4, BORDER),
231     ]))
232     return t
233
234

```

[illegible]

```

for i, (_, c) in enumerate(tags):
329     ts.append(("BACKGROUND", (i,0), (i,0), c))
330     tgt.setStyle(TableStyle(ts))
331     story.append(tgt)
332     story.append(Spacer(1, 8*mm))
333
334 # =====
335 # 01 – RESUMO EXECUTIVO
336 # =====
337 for e in sec("01 · Resumo", "Executivo – Visão Geral do Projeto", styles):
338     story.append(e)
339
340 story.append(Paragraph(
341     "Este documento descreve, de forma completa e rastreável, toda a stack tecnológica, "
342     "o prompt exato e o código-fonte utilizado para gerar o PDF ultra premium do "
343     "<b>Plano de Aula – Introdução à Análise de Dados Corporais com Inteligência Artificial "
344     "na Educação Física Escolar</b>. O objetivo é garantir reprodutibilidade total, "
345     "transparência metodológica e documentação de engenharia de prompt.",
346     styles["body"]))
347
348 # Pipeline visual
349 pipeline_items = [
350     ("1", GOLD, "PROMPT", "Engenharia de prompt\nmultidisciplinar"),
351     ("2", ACCENT_PURPLE, "CLAUDE", "Processamento\nClaude Opus 4.6"),
352     ("3", TEAL, "PYTHON", "Geração de código\nReportLab"),
353     ("4", NAVY, "PDF", "Renderização\nA4 Premium"),
354 ]
355 pip_row = []
356 for num, color, label, desc in pipeline_items:
357     pip_row.append([
358         Paragraph(label, ParagraphStyle("pl", fontName="Helvetica-Bold",
359             fontSize=9, textColor=WHITE, alignment=TA_CENTER)),
360         Paragraph(desc, ParagraphStyle("pd", fontName="Helvetica",
361             fontSize=7, leading=10, textColor=TEXT_MED, alignment=TA_CENTER)),
362     ])
363 pip_data = [
364     [pip_row[0][0], Paragraph(">>>", ParagraphStyle("a", fontName="Helvetica-Bold",
365         fontSize=12, textColor=GOLD, alignment=TA_CENTER))],
366     [pip_row[1][0], Paragraph(">>>", ParagraphStyle("a2", fontName="Helvetica-Bold",
367         fontSize=12, textColor=GOLD, alignment=TA_CENTER))],
368     [pip_row[2][0], Paragraph(">>>", ParagraphStyle("a3", fontName="Helvetica-Bold",
369         fontSize=12, textColor=GOLD, alignment=TA_CENTER))],
370     [pip_row[3][0],
371     [pip_row[0][1], Paragraph("", styles["body"])],
372     [pip_row[1][1], Paragraph("", styles["body"])],
373     [pip_row[2][1], Paragraph("", styles["body"])],
374     [pip_row[3][1]],
375 ]
376 arrow_w = 1.2*cm
377 box_w = (COL_W - 3*arrow_w) / 4
378 pip_t = Table(pip_data, colWidths=[box_w, arrow_w, box_w, arrow_w, box_w, arrow_w, box_w])
379 pip_styles = [
380     ("ALIGN", (0,0), (-1,-1), "CENTER"),
381     ("VALIGN", (0,0), (-1,-1), "MIDDLE"),
382     ("TOPPADDING", (0,0), (-1,-1), 6),
383     ("BOTTOPPADDING", (0,0), (-1,-1), 6),
384 ]
385 for i, (_, color, _, _) in enumerate(pipeline_items):
386     col = i * 2
387     pip_styles.append(("BACKGROUND", (col,0), (col,0), color))
388 pip_t.setStyle(TableStyle(pip_styles))
389 story.append(Spacer(1, 4))
390 story.append(pip_t)
391 story.append(Spacer(1, 2))
392 story.append(Paragraph("Pipeline de geração: do prompt ao PDF final.", styles["caption"]))
393 story.append(Spacer(1, 6*mm))
394
395 # =====
396 # 02 – STACK TECNOLÓGICA COMPLETA
397 # =====
398 for e in sec("02 · Stack", "Tecnológica Completa – Todas as Camadas", styles):
399     story.append(e)
400
401 techs = [
402     ("AI", ACCENT_PURPLE, "Claude AI – Opus 4.6 (Anthropic)",
403     "Modelo de linguagem de grande porte (LLM) utilizado como motor de raciocínio, "
404     "engenharia de prompt, estruturação pedagógica, geração de código Python e "
405     "design de layout do PDF. Operando em modo agêntico via Claude Agent SDK.",
406     "opus-4.6"),
407     ("PY", NAVY, "Python 3.10 – Linguagem Principal",
408     "Linguagem de programação integrada para toda a lógica de geração do PDF. "
409     "Executada em ambiente Linux (Ubuntu 22) via sandbox seguro do Cowork Mode.",
410     "3.10"),
411     ("RL", TEAL, "ReportLab 4.4.10 – Motor de PDF",
412     "Biblioteca Python de criação programática de PDFs. Utilizado o módulo Platypus "
413     "(Page Layout and Typography Using Scripts) para layout automático com flowables, "
414     "e Canvas para header/footer customizados com faixas coloridas.",
415     "4.4.10"),
416     ("CW", ACCENT_ORANGE, "Cowork Mode – Claude Desktop App",
417     "Ambiente de execução integrado ao Claude Desktop. Fornece VM Linux leve com "
418     "acesso ao sistema de arquivos, terminal bash, e ferramentas de leitura/escrita. "
419     "Permite execução de código Python e entrega de arquivos ao usuário.",
420     "2026").

```



```

421 ("SDK", NAVY_MED, "Claude Agent SDK – Orquestração",
422 "Framework de agentes da Anthropic que permite execução autônoma de tarefas "
423 "multi-etapa: leitura de skills, instalação de pacotes, escrita de código, "
424 "execução e verificação – tudo em sequência agêntica.",
425 "latest"),
426 ("SK", GREEN_OK, "Skills System – PDF Skill",
427 "Sistema de skills do Cowork Mode. O skill 'pdf' foi consultado previamente "
428 "para obter best practices de geração de PDF com ReportLab (Platypus + Canvas), "
429 "incluindo orientações sobre fontes, subscripts e layout.",
430 "v1.0"),
431 ("PIP", HexColor("#7B3F00"), "pip – Gerenciador de Pacotes",
432 "Utilizado para instalar a biblioteca ReportLab no ambiente de execução: "
433 "pip install reportlab --break-system-packages",
434 "pip3"),
435 ("OS", HexColor("#6B7280"), "Ubuntu 22 (Linux) – Sistema Operacional",
436 "VM Linux leve que serve como sandbox de execução segura. Fornece isolamento "
437 "do sistema host enquanto permite operações de arquivo e execução de scripts.",
438 "22.04"),
439 ]
440
441 for sigla, bg, nome, desc, versao in techs:
442     story.append(tech_row(sigla, bg, nome, desc, versao, styles))
443     story.append(Spacer(1, 2))
444
445 story.append(Spacer(1, 6*mm))
446
447 # ████████████████████████████████████████████████████████████████████████████
448 # 03 – PROMPT EXATO UTILIZADO
449 # ████████████████████████████████████████████████████████████████████████████
450 for e in sec("03 · Prompt", "Exato Fornecido pelo Usuário (Integral)", styles):
451     story.append(e)
452
453     story.append(Paragraph(
454         "Abaixo está o prompt <b>integral e exato</b>, sem nenhuma modificação, "
455         "conforme enviado pelo usuário na interface do Cowork Mode. "
456         "Este prompt acionou toda a cadeia de geração documentada neste relatório.",
457         styles["body"]))
458     story.append(Spacer(1, 3))
459
460     prompt_text = ""
461 Vou estruturar como voce pediu:
462 * Especialista em Educacao Fisica Escolar
463 * Especialista em Inteligencia Artificial
464 * Especialista em Business Intelligence
465 * Especialista em Legislacao Educacional Brasileira
466 * Especialista em Neurociencia e Aprendizagem
467 * Especialista em Metodologia Cientifica
468 * Consultor em Politicas Publicas Educacionais
469
470 E vamos fundamentar em:
471 * Lei de Diretrizes e Bases da Educacao Nacional
472 * Base Nacional Comum Curricular
473 * Lei Geral de Protecao de Dados Pessoais
474 * Ministerio da Educacao
475 * Conselho Nacional de Educacao
476
477 PLANO DE AULA - VERSAO INSTITUCIONAL E CIENTIFICA
478 Introducao a Analise de Dados Corporais com Inteligencia
479 Artificial na Educacao Fisica Escolar
480
481 1. Fundamentacao Legal e Normativa
482 Este plano esta alinhado com:
483 LDB - Lei de Diretrizes e Bases da Educacao Nacional
484 Art. 2 e Art. 35 - Formacao integral do educando.
485 Base Nacional Comum Curricular
486 Competencia Geral 5 - Cultura Digital
487 Competencia Geral 2 - Pensamento Cientifico, Critico e Criativo
488 Lei Geral de Protecao de Dados Pessoais
489 Principios de finalidade, necessidade e transparencia no uso
490 de dados.
491
492 2. Fundamentacao Cientifica
493 Baseado em evidencias de:
494 * Alfabetizacao em dados na educacao basica
495 * Data literacy como competencia do seculo XXI
496 * Aprendizagem baseada em problemas
497 * Neurociencia aplicada a aprendizagem ativa
498 * Pensamento analitico e tomada de decisao baseada em dados
499
500 Estudos internacionais indicam que introducao precoce ao
501 raciocinio baseado em dados aumenta:
502 * Capacidade de interpretacao critica
503 * Autonomia cognitiva
504 * Transferencia interdisciplinar
505
506 3. Justificativa Pedagogica
507 A Educacao Fisica, tradicionalmente centrada na pratica
508 corporal, pode atuar como campo interdisciplinar para:
509 * Introduzir conceitos de coleta de dados
510 * Desenvolver raciocinio quantitativo
511 * Trabalhar etica digital
512 * Promover consciencia corporal com base cientifica
513

```

```
514 4. Objetivos de Aprendizagem (Mensuraveis)
515 Ao final da aula, o estudante devera ser capaz de:
516 1. Identificar variaveis fisiologicas basicas (tempo, FC, passos)
517 2. Interpretar dados estruturados em tabela
518 3. Comparar desempenhos com base em criterios objetivos
519 4. Explicar de forma simples como a IA identifica padroes
520 5. Discutir implicacoes eticas do uso de dados
521
522 5. Estrutura Metodologica (Modelo Ativo)
523 5.1 Situacao-Problema
524 Dois alunos percorrem a mesma distancia. Quem apresentou
525 melhor desempenho?
526 Variaveis analisadas: Tempo, Frequencia cardiaca media,
527 Numero de passos
528
529 5.2 Conceituacao Tecnica Simplificada
530 Equipe de IA explica:
531 * Dados estruturados
532 * Identificacao de padroes
533 * Diferenca entre esforco e eficiencia
534 * Conceito introdutorio de algoritmo
535 Equipe de BI explica:
536 Fluxo decisorio: Dados > Organizacao > Visualizacao >
537 Analise > Decisao
538
539 5.3 Atividade Pratica Orientada
540 Tabela ficticia:
541 Aluno | Tempo | FC Media | Passos
542 A | 12 min | 165 bpm | 1800
543 B | 10 min | 178 bpm | 2100
544 C | 11 min | 155 bpm | 1700
545
546 Perguntas guiadas:
547 * Quem teve maior esforco?
548 * Quem teve melhor eficiencia?
549 * Que variavel pesa mais?
550
551 5.4 Discussao Etica
552 Base legal: Lei Geral de Protecao de Dados Pessoais
553 Reflexoes: Consentimento, Armazenamento, Finalidade pedagogica
554
555 6. Diferenciacao Pedagogica Estruturada
556 Aluno com dificuldade: Uso de graficos visuais, Explicacao
557 oral mediada
558 Aluno avancado: Desafio: propor modelo simples de pontuacao
559 ponderada
560 Perfil analitico: Calculo comparativo
561 Perfil verbal: Debate argumentativo
562
563 7. Avaliacao (Modelo Rubricado)
564 Criterio | Excelente | Adequado | Inicial
565 Interpretacao | Analisa criticamente | Interpreta parcial | Dificuldade
566 Argumentacao | Fundamenta com clareza | Argumenta superficial | Nao argumenta
567 Participacao | Ativa | Moderada | Baixa
568
569 8. Uso do Ecossistema IA
570 A equipe tecnica utilizou IA para:
571 * Estruturacao metodologica
572 * Simulacao de dados
573 * Diferenciacao pedagogica
574 * Organizacao textual
575 Validacao final realizada pelo docente.
576
577 9. Feedback Multidisciplinar
578 Educacao Fisica: Integra ciencia do movimento com cultura digital
579 IA: Introducao correta ao conceito de reconhecimento de padroes
580 BI: Fluxo de decisao estruturado adequadamente
581 Juridico: Alinhado a LGPD
582 Neurociencia: Metodologia ativa favorece consolidacao
583
584 10. Conclusao Institucional
585 Este plano:
586 * Atende as diretrizes da Base Nacional Comum Curricular
587 * Esta alinhado a Lei de Diretrizes e Bases da Educacao Nacional
588 * Respeita principios da Lei Geral de Protecao de Dados Pessoais
589 * Introduz alfabetizacao em dados no Ensino Fundamental
590 * Desenvolve pensamento critico e cultura digital
591
592 crie um pdf ultra premium de elite, com rodape e cabecalho
593 com faixa, frase dentro do cabecalho (titulo)
594 organize como um designers"""
595
596 for el in code_block("PROMPT INTEGRAL DO USUARIO - Cowork Mode", prompt_text, styles):
597     story.append(el)
598
599 story.append(Spacer(1, 6*mm))
600
601 #
602 # 04 - DECISÖES DE DESIGN
603 #
604 for e in sec("04 - Design", "Decisões de Design e Arquitetura Visual", styles):
605     story.append(e)
606
```

```

design_decisions = [
    ("Paleta de Cores",
     "Navy #0D1B2A (institucional), Teal #0E7C7B (accent primário), "
     "Gold #C9A227 (accent secundário), #F4F7FB (fundo leve). "
     "Paleta de 3 cores primárias + neutros para máxima legibilidade e hierarquia."),
    ("Tipografia",
     "Helvetica (sans-serif) em 4 pesos: Bold para títulos, Regular para corpo, "
     "Oblique para captions, Courier para código. Tamanhos: 18pt (título), "
     "13pt (seções), 9pt (corpo), 8pt (tabelas), 7.5pt (captions/labels)."),
    ("Header (Cabeçalho)",
     "Faixa tripla: Navy 22mm (principal) + Gold 2mm + Teal 1.5mm. "
     "Ornamentos laterais: Gold esquerdo + Teal direito (6mm cada). "
     "Linha decorativa interna Gold Light. Texto centralizado em branco."),
    ("Footer (Rodapé)",
     "Espelhamento do header: Teal 1.5mm + Gold 1.5mm + Navy 14.5mm. "
     "Ornamentos laterais idênticos. 3 zonas: info esquerda, "
     "página central, data direita."),
    ("Layout de Seções",
     "Cada seção: Label uppercase (TEAL, 7.5pt) + Título bold (NAVY, 13pt) "
     "+ divider horizontal. Numeração sequencial (01, 02...). "
     "Espaçamento consistente via Spacer."),
    ("Tabelas",
     "Header Navy com texto branco. Linhas alternadas (#E8F0F8 / #FFFFFF). "
     "Borda lateral Teal (linebefore). Box externo 0.8pt. "
     "Badges coloridos por área/categoria."),
    ("Blocos de Destaque",
     "Caixas com borda lateral colorida (3pt), fundo leve, "
     "para situação-problema e callouts importantes."),
    ("Formato Final",
     "A4 (595.28 x 841.89 pts). Margens: L/R 2cm, Top 3.2cm, Bottom 2.8cm. "
     "6 páginas. 22KB. Metadata: título, autor, assunto, criador."),
]

for titulo, desc in design_decisions:
    row = [
        Paragraph(titulo, ParagraphStyle("dt", fontName="Helvetica-Bold",
                                         fontSize=8.5, textColor=NAVY, spaceAfter=1)),
        Paragraph(desc, ParagraphStyle("dd", fontName="Helvetica",
                                       fontSize=8, leading=12, textColor=TEXT_MED, alignment=TA_JUSTIFY)),
    ]
    t = Table(row, colWidths=[3.2*cm, COL_W - 3.2*cm])
    t.setStyle(TableStyle([
        ("BACKGROUND", (0,0), (0,0), LIGHT_BG2),
        ("BACKGROUND", (1,0), (1,0), LIGHT_BG),
        ("VALIGN", (0,0), (-1,-1), "TOP"),
        ("LEFTPADDING", (0,0), (-1,-1), 8),
        ("RIGHTPADDING", (0,0), (-1,-1), 8),
        ("TOPPADDING", (0,0), (-1,-1), 6),
        ("BOTTOMPADDING", (0,0), (-1,-1), 6),
        ("LINEBELOW", (0,0), (-1,-1), 0.3, BORDER),
        ("LINEAFTER", (0,0), (0,-1), 0.8, TEAL),
    ]))
    story.append(t)

story.append(Spacer(1, 6*mm))

# =====
# 05 – MÓDULOS REPORTLAB UTILIZADOS
# =====
for e in sec("05 · ReportLab", "Módulos e Classes Utilizados", styles):
    story.append(e)

modules = [
    ("reportlab.lib.pagesizes", "A4", "Dimensões da página (595.28 x 841.89 pts)",),
    ("reportlab.lib.colors", "HexColor", "Sistema de cores hexadecimal customizado",),
    ("reportlab.lib.units", "mm, cm", "Unidades métricas para posicionamento",),
    ("reportlab.lib.styles", "ParagraphStyle", "Estilos tipográficos customizados",),
    ("reportlab.lib.enums", "TA_LEFT, TA_CENTER, ...", "Alinhamento de texto",),
    ("reportlab.platypus", "SimpleDocTemplate", "Motor de layout automático de páginas",),
    ("reportlab.platypus", "Paragraph", "Elemento de texto com formatação rica",),
    ("reportlab.platypus", "Table, TableStyle", "Tabelas com estilização completa",),
    ("reportlab.platypus", "Spacer", "Controle de espaçamento vertical",),
    ("reportlab.platypus", "HRFlowable", "Linhas horizontais decorativas",),
    ("reportlab.platypus", "Flowable (base class)", "Classe base para ColorBox customizado",),
    ("reportlab.pdfgen", "canvas.Canvas", "Desenho direto para header/footer",),
]

mod_header = [Paragraph(h, styles["tbl_header"])
               for h in ["MÓDULO", "CLASSE / IMPORT", "FINALIDADE"]]
mod_rows = [mod_header]
for mod, cls, fin in modules:
    mod_rows.append([
        Paragraph(mod, ParagraphStyle("mm", fontName="Courier",
                                       fontSize=6.5, leading=9, textColor=TEAL)),
        Paragraph(cls, ParagraphStyle("mc", fontName="Courier-Bold",
                                       fontSize=7, leading=9, textColor=NAVY)),
        Paragraph(fin, ParagraphStyle("mf", fontName="Helvetica",
                                       fontSize=7.8, leading=10, textColor=TEXT_MED)),
    ])

mod_t = Table(mod_rows, colWidths=[4.2*cm, 4.3*cm, COL_W - 8.5*cm])
mod_ts = [
    ("BACKGROUND", (0,0), (-1,0), NAVY),

```

```

0 ("ROWBACKGROUND", (0,0), (-1,-1), [STRIPE_1, STRIPE_2]),
701 ("ALIGN", (0,0), (-1,0), "CENTER"),
702 ("VALIGN", (0,0), (-1,-1), "MIDDLE"),
703 ("LEFTPADDING", (0,0), (-1,-1), 6),
704 ("RIGHTPADDING", (0,0), (-1,-1), 6),
705 ("TOPPADDING", (0,0), (-1,-1), 5),
706 ("BOTTOMPADDING", (0,0), (-1,-1), 5),
707 ("LINEBELOW", (0,0), (-1,-1), 0.3, BORDER),
708 ("BOX", (0,0), (-1,-1), 0.8, NAVY),
709 ]
710 mod_t.setStyle(TableStyle(mod_ts))
711 story.append(mod_t)
712 story.append(Spacer(1, 6*mm))
713
714 # ████████████████████████████████████████████████████████████████████████████
715 # 06 - FLUXO DE EXECUÇÃO
716 # ████████████████████████████████████████████████████████████████████████████
717 for e in sec("06 · Fluxo", "de Execução Agêntico – Passo a Passo", styles):
718     story.append(e)
719
720 story.append(Paragraph(
721     "O processo de geração seguiu um fluxo agêntico automatizado, "
722     "onde o Claude AI executou cada etapa de forma autônoma e sequencial:",
723     styles["body"]))
724
725 steps = [
726     ("01", TEAL, "Leitura do Skill PDF",
727         "Claude leu o arquivo SKILL.md do skill 'pdf' para obter best practices de geração "
728         "com ReportLab, incluindo orientações sobre Platypus, Canvas e fontes."),
729     ("02", GOLD, "Análise do Prompt",
730         "Interpretação semântica do prompt do usuário: identificação das 10 seções do plano, "
731         "requisitos de design (cabeçalho com faixa, rodapé, nível editorial) e público-alvo."),
732     ("03", NAVY, "Instalação de Dependências",
733         "Execução: pip install reportlab --break-system-packages. "
734         "Verificação: import reportlab; print(reportlab.Version) => 4.4.10"),
735     ("04", ACCENT_PURPLE, "Geração do Código Python",
736         "Escrita de ~700 linhas de Python com: paleta de cores (9 cores), "
737         "sistema de estilos (15+ ParagraphStyles), função header/footer customizada, "
738         "9 seções de conteúdo com tabelas, badges e blocos de destaque."),
739     ("05", NAVY_MED, "Execução e Renderização",
740         "Execução do script: python3 plano_aula_premium.py. "
741         "ReportLab processou o story (lista de flowables) e renderizou 6 páginas A4."),
742     ("06", GREEN_OK, "Verificação e Entrega",
743         "Validação: leitura com pypdf (6 páginas, metadados OK, 22KB). "
744         "Arquivo salvo em /mnt/outputs/ e link computer:// fornecido ao usuário."),
745 ]
746
747 for num, color, titulo, desc in steps:
748     row = [[
749         Paragraph(num, ParagraphStyle("sn", fontName="Helvetica-Bold",
750             fontSize=10, textColor=WHITE, alignment=TA_CENTER)),
751         [Paragraph(titulo, ParagraphStyle("stt", fontName="Helvetica-Bold",
752             fontSize=8.5, textColor=NAVY, spaceAfter=1)),
753          Paragraph(desc, ParagraphStyle("sd", fontName="Helvetica",
754             fontSize=8, leading=12, textColor=TEXT_MED, alignment=TA_JUSTIFY))],
755     ]
756     t = Table(row, colWidths=[1.0*cm, COL_W - 1.0*cm])
757     t.setStyle(TableStyle([
758         ("BACKGROUND", (0,0), (0,0), color),
759         ("BACKGROUND", (1,0), (1,0), LIGHT_BG),
760         ("VALIGN", (0,0), (-1,-1), "MIDDLE"),
761         ("LEFTPADDING", (0,0), (-1,-1), 8),
762         ("RIGHTPADDING", (0,0), (-1,-1), 8),
763         ("TOPPADDING", (0,0), (-1,-1), 7),
764         ("BOTTOMPADDING", (0,0), (-1,-1), 7),
765         ("LINEBELOW", (0,0), (-1,-1), 0.4, BORDER),
766     ]))
767     story.append(t)
768     story.append(Spacer(1, 2))
769
770 story.append(Spacer(1, 6*mm))
771
772 # ████████████████████████████████████████████████████████████████████████████
773 # 07 - COMANDOS BASH EXECUTADOS
774 # ████████████████████████████████████████████████████████████████████████████
775 for e in sec("07 · Comandos", "Bash Executados no Terminal", styles):
776     story.append(e)
777
778     bash_code = """# 1. Instalacao da biblioteca ReportLab
779 pip install reportlab --break-system-packages -q
780
781 # 2. Verificacao da instalacao
782 python3 -c "import reportlab; print('OK', reportlab.Version)"
783 # Saida: OK 4.4.10
784
785 # 3. Execucao do script de geracao do PDF
786 python3 plano_aula_premium.py
787 # Saida: PDF gerado: /sessions/.../mnt/outputs/Plano_Aula_IA_EF_Premium.pdf
788
789 # 4. Verificacao do PDF gerado
790 ls -lh /sessions/.../mnt/outputs/Plano_Aula_IA_EF_Premium.pdf
791 # Saida: 22K Feb 25 00:50 Plano_Aula_IA_EF_Premium.pdf
792

```

[illegible]

```
886     styles = S()
887     story = build(styles)
888     doc.build(story, onFirstPage=draw_header_footer, onLaterPages=draw_header_footer)
889     print(f"PDF gerado: {OUTPUT}")
890
891 if __name__ == "__main__":
892     main()
893
```