



## Relatório de Análise de Dados: Insights para a Estratégia Bellabeat

### 1. Introdução

Este relatório técnico detalha o processo de análise de dados de dispositivos inteligentes não-Bellabeat, com o objetivo de extrair insights valiosos para aprimorar a estratégia de marketing da Bellabeat, focando especificamente no produto Bellabeat Leaf. A análise abrange padrões de atividade física, sono e peso, buscando identificar tendências de comportamento do consumidor que possam ser aplicadas para otimizar o engajamento dos usuários e o desenvolvimento de produtos.

### 2. Visão Geral do Código e Metodologia

O projeto foi desenvolvido em **R**, utilizando o ambiente **RStudio** e pacotes do **tidyverse** para manipulação e visualização de dados. A metodologia seguiu as etapas de carregamento, limpeza, transformação, fusão de dados, análise descritiva, segmentação de usuários e geração de visualizações.

#### 2.1. Carregamento de Bibliotecas e Configuração do Ambiente

Esta seção inicial carrega as bibliotecas necessárias para a análise e configura o diretório de trabalho para garantir que todos os arquivos sejam localizados corretamente.

```
1 # Carregamento das Bibliotecas
2 library(tidyverse)
3 library(lubridate)
4 library(ggplot2)
5 library(dplyr)
6 library(readr)
7
8 # (Melhoria: Definir diretório de trabalho relativo ao script)
9 # Esta linha garante que o RStudio defina o diretório de trabalho para a localização do script.
10 # Isso é crucial para que os caminhos relativos funcionem corretamente.
11 # Certifique-se de que o pacote 'rstudioapi' esteja instalado (install.packages('rstudioapi'))
12 # Se estiver rodando fora do RStudio, você precisará definir o diretório de trabalho manualmente.
13
14 if (requireNamespace("rstudioapi", quietly = TRUE)) {
15   setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
16 }
```

#### 2.2. Funções de Carregamento e Limpeza de Dados

Foram criadas funções modularizadas para carregar e realizar a limpeza inicial de cada dataset (atividade diária, sono e peso). Isso inclui renomear colunas, converter tipos de dados (especialmente datas) e remover duplicatas, garantindo a consistência e qualidade dos dados antes da fusão.

```
18 # 1 - FUNÇÕES DE CARREGAMENTO E LIMPEZA DE DADOS (DEFINIÇÕES)
19
20 # Função para carregar e limpar dados de atividade diária
21
22 load_and_clean_daily_activity <- function(file_path) {
23   daily_activity <- read_csv(file_path)
24
25   # Renomear colunas para clareza e consistência
26
27   daily_activity <- daily_activity %>%
28     rename(date = ActivityDate,
29            total_steps = TotalSteps,
30            calories = Calories)
31
32   # Converter 'Id' para caractere e 'date' para o formato de data
33
34   daily_activity$Id <- as.character(daily_activity$Id)
35   daily_activity$date <- mdy(daily_activity$date)
36
37   # Criar uma nova coluna para o dia da semana
38   daily_activity$day_of_week <- weekdays(daily_activity$date, abbreviate = FALSE)
39 }
```



```
39
40 # verificar e remover duplicatas (Melhoria: Tratamento explícito de duplicatas)
41 if (sum(duplicated(daily_activity)) > 0) {
42   message(paste("Removendo", sum(duplicated(daily_activity)), "linhas duplicadas em daily_activity."))
43   daily_activity <- daily_activity[!duplicated(daily_activity), ]
44 }
45
46 return(daily_activity)
47 }
48
49 # Função para carregar e limpar dados de sono
50 load_and_clean_sleep_data <- function(file_path) {
51   sleep_day <- read_csv(file_path)
52
53   # Renomear e limpar os dados de sono
54   sleep_day <- sleep_day %>%
55     rename(sleep_date = SleepDay,
56            total_sleep_minutes = TotalMinutesAsleep,
57            total_time_in_bed = TotalTimeInBed) %>%
58     mutate(sleep_date = mdy_hms(sleep_date)) %>%
59     mutate(sleep_date = as.Date(sleep_date)) # Converter para formato de data
60
61   # Converter 'Id' para caractere
62   sleep_day$Id <- as.character(sleep_day$Id)
63
64   # verificar e remover duplicatas (Melhoria: Tratamento explícito de duplicatas)
65   if (sum(duplicated(sleep_day)) > 0) {
66     message(paste("Removendo", sum(duplicated(sleep_day)), "linhas duplicadas em sleep_day."))
67     sleep_day <- sleep_day[!duplicated(sleep_day), ]
68   }
69 }
```

```
69
70 return(sleep_day)
71 }
72
73 # Função para carregar e limpar dados de peso
74 load_and_clean_weight_data <- function(file_path) {
75   weight_info <- read_csv(file_path)
76
77   # Renomear e limpar os dados de peso
78   weight_info <- weight_info %>%
79     rename(date = Date,
80            weight_kg = weightkg) %>%
81     mutate(date = mdy_hms(date)) %>%
82     mutate(date = as.Date(date)) # Converter para formato de data
83
84   # Converter 'Id' para caractere
85   weight_info$Id <- as.character(weight_info$Id)
86
87   # verificar e remover duplicatas (Melhoria: Tratamento explícito de duplicatas)
88   if (sum(duplicated(weight_info)) > 0) {
89     message(paste("Removendo", sum(duplicated(weight_info)), "linhas duplicadas em weight_info."))
90     weight_info <- weight_info[!duplicated(weight_info), ]
91   }
92
93   return(weight_info)
94 }
95 }
```

### 2.3. Carregamento e Fusão dos Datasets

Após a definição das funções, os datasets são carregados utilizando caminhos relativos ajustados à estrutura do projeto e, em seguida, mesclados. A fusão é realizada com (*all.x = TRUE*) para garantir que todas as entradas de atividade sejam mantidas, mesmo que não haja dados correspondentes de sono ou peso para todas as datas/IDs. É importante notar que o dataset de peso possui uma amostra limitada, o que exige cautela na interpretação das análises que o envolvem.

```
96 # CHAMADAS DAS FUNÇÕES (AGORA QUE ELAS JÁ FORAM DEFINIDAS)
97 # Carregar os datasets usando caminhos relativos (Melhoria: Caminhos Relativos ajustados)
98 # O script está em 'Rproj01/03_Scripts', e os CSVs em 'Rproj01/02_Dates'
99 # Então, para ir de '03_Scripts' para '02_Dates', precisamos subir um nível (..) e descer para '02_Dates'
100 daily_activity <- load_and_clean_daily_activity("../02_Dates/dailyActivity_merged.csv")
101 sleep_day <- load_and_clean_sleep_data("../02_Dates/sleepDay_merged.csv")
102 weight_info <- load_and_clean_weight_data("../02_Dates/weightLogInfo_merged.csv")
103
104 # Get a quick overview of each dataset
105 glimpse(daily_activity)
106 glimpse(sleep_day)
107 glimpse(weight_info)
108
109 # Verificar IDs únicos e duplicatas (já feito nas funções, mas mantido para verificação)
110 n_distinct(daily_activity$Id)
111 sum(duplicated(daily_activity))
112
113 # Verificar valores ausentes (Melhoria: Documentar a decisão de não imputar NAs para esta análise)
114 # Para esta análise exploratória, optamos por não imputar valores ausentes,
115 # pois a remoção de linhas com NAs pode reduzir significativamente o dataset,
116 # especialmente o de peso. As análises serão feitas com os dados disponíveis.
117 sum(is.na(daily_activity))
118 sum(is.na(sleep_day))
119 sum(is.na(weight_info)) # O dataset de peso tem NAs na coluna 'Fat', que não será usada diretamente.
120
```



```
121 # Fusão dos datasets para uma análise mais abrangente
122 # (Melhoria: Reconhecer a limitação do dataset de peso)
123 # A fusão resultará em um dataset menor devido ao número limitado de registros de peso.
124 # As análises envolvendo peso devem ser interpretadas com cautela.
125 merged_data <- merge(daily_activity, sleep_day, by.x = c("id", "date"), by.y = c("id", "sleep_date"), all.x = TRUE)
126 final_merged_data <- merge(merged_data, weight_info, by = c("id", "date"), all.x = TRUE) # Usar all.x para manter a
127
128 glimpse(final_merged_data)
129
```

## 2.4. Análise Descritiva e Segmentação de Usuários

Esta etapa foca na obtenção de estatísticas descritivas para as variáveis-chave de atividade e sono, além de calcular uma matriz de correlação para identificar relações entre as variáveis. Uma segmentação de usuários baseada em seus níveis de atividade média também é realizada para permitir análises mais direcionadas.

```
130 # --- 2. Análise Descritiva e Aprofundamento (Melhorias: Sono, Correlação, Segmentação) ---
131
132 # Estatísticas descritivas para colunas chave de atividade
133 cat("\nEstatísticas Descritivas de Atividade:\n")
134 daily_activity %>%
135   select(total_steps, calories, VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, SedentaryMinutes) %>%
136   summary()
137
138 # Média de minutos gastos em cada nível de atividade
139 cat("\nMédia de Minutos por Nível de Atividade:\n")
140 daily_activity %>%
141   summarise(
142     avg_very_active = mean(VeryActiveMinutes, na.rm = TRUE),
143     avg_fairly_active = mean(FairlyActiveMinutes, na.rm = TRUE),
144     avg_lightly_active = mean(LightlyActiveMinutes, na.rm = TRUE),
145     avg_sedentary = mean(SedentaryMinutes, na.rm = TRUE)
146   ) %>%
147   print()
148
149 # (Melhoria: Estatísticas Descritivas do Sono)
150 cat("\nEstatísticas Descritivas de Sono:\n")
151 sleep_day %>%
152   select(total_sleep_minutes, total_time_in_bed) %>%
153   summary()
154
```

```
155 # (Melhoria: Análise de Correlação)
156 # Calculando a matriz de correlação para variáveis numéricas relevantes
157 # Removendo NAs para o cálculo da correlação
158 correlation_data <- final_merged_data %>%
159   select(total_steps, calories, VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, SedentaryMinutes, tot
160   drop_na()
161
162 if (nrow(correlation_data) > 1) { # Verificar se há dados suficientes para correlação
163   correlation_matrix <- cor(correlation_data)
164   cat("\nMatriz de Correlação:\n")
165   print(round(correlation_matrix, 2))
166 } else {
167   cat("\nNão há dados suficientes para calcular a matriz de correlação após remover NAs.\n")
168 }
169
170 # (Melhoria: Segmentação de Usuários por Nível de Atividade)
171 # Definindo níveis de atividade com base na média de minutos ativos
172 user_activity_summary <- daily_activity %>%
173   group_by(id) %>%
174   summarise(
175     avg_total_steps = mean(total_steps, na.rm = TRUE),
176     avg_very_active_minutes = mean(VeryActiveMinutes, na.rm = TRUE),
177     avg_fairly_active_minutes = mean(FairlyActiveMinutes, na.rm = TRUE),
178     avg_lightly_active_minutes = mean(LightlyActiveMinutes, na.rm = TRUE),
179     avg_sedentary_minutes = mean(SedentaryMinutes, na.rm = TRUE),
180     total_days_logged = n_distinct(date)
181   ) %>%
182
```

```
182   mutate(
183     activity_level = case_when(
184       avg_very_active_minutes > 30 ~ "Muito Ativo",
185       avg_fairly_active_minutes > 20 ~ "Ativo Moderado",
186       avg_lightly_active_minutes > 150 ~ "Levemente Ativo",
187       TRUE ~ "Sedentário"
188     )
189   )
190
191 cat("\nResumo da Segmentação de Usuários por Nível de Atividade:\n")
192 user_activity_summary %>%
193   group_by(activity_level) %>%
194   summarise(count = n(), avg_steps = mean(avg_total_steps)) %>%
195   print()
196
197 # Juntar com os dados finais para análises segmentadas
198 final_merged_data_segmented <- merge(final_merged_data, user_activity_summary %>%
199   select(id, activity_level, total_days_logged),
200   by = "id", all.x = TRUE)
201
```



## 2.5. Visualizações

Esta seção gera os gráficos que ilustram as principais tendências e relações nos dados. Um tema visual consistente é aplicado para garantir a clareza e profissionalismo das visualizações.

```
202 # --- 3. Visualizações (Melhorias: Consistência, Títulos, Gráficos Adicionais) ---
203
204 # Definir um tema consistente para os gráficos (Melhoria: Consistência Visual)
205 my_theme <- theme_minimal() +
206   theme(
207     plot.title = element_text(hjust = 0.5, face = "bold"),
208     axis.title = element_text(face = "bold"),
209     legend.title = element_text(face = "bold"),
210     axis.text.x = element_text(angle = 45, hjust = 1)
211   )
212
213 # Gráfico 1: Scatter plot de Total Steps vs. Calories (Melhoria: Coeficiente de correlação)
214 plot_steps_calories <- ggplot(data = daily_activity, aes(x = total_steps, y = calories)) +
215   geom_point(color = "#FF6347", alpha = 0.6) +
216   geom_smooth(method = "lm", color = "darkblue", se = FALSE) +
217   labs(title = "Relação entre Passos Totais e Calorias Queimadas",
218        x = "Total de Passos",
219        y = "Calorias Queimadas") +
220   my_theme
221 print(plot_steps_calories)
222
223 # Gráfico 2: Bar chart de Average Steps by Day of the Week
224 plot_avg_steps_weekday <- daily_activity %>%
225   group_by(day_of_week) %>%
226   summarise(avg_steps = mean(total_steps, na.rm = TRUE)) %>%
227   mutate(day_of_week = factor(day_of_week, levels = c("domingo", "segunda-feira", "terça-feira", "quarta-feira", "quinta-feira", "sexta-feira", "sábado")))
228 ggplot(aes(x = day_of_week, y = avg_steps, fill = day_of_week)) +
229   geom_bar(stat = "identity") +
230   # LINHA ADICIONADA PARA OS RÓTULOS
231   geom_text(aes(label = round(avg_steps, 0)), vjust = -0.5, color = "black", size = 3.5) +
232   labs(title = "Média de Passos por Dia da Semana",
233        x = "Dia da Semana",
234        y = "Média de Passos") +
235   my_theme +
236   theme(legend.position = "none")
237 print(plot_avg_steps_weekday)
238
239 # Gráfico 3: Scatter plot de Total Steps vs. Weight (Melhoria: Nota sobre limitação da amostra)
240 plot_steps_weight <- ggplot(data = final_merged_data, aes(x = total_steps, y = weight_kg)) +
241   geom_point(color = "#6A5ACD", alpha = 0.7) +
242   labs(title = "Relação entre Peso e Total de Passos (Amostra Limitada)",
243        x = "Total de Passos",
244        y = "Peso (kg)",
245        caption = "*Atenção: Dados de peso limitados, interpretar com cautela.") +
246   my_theme
247 print(plot_steps_weight)
248
249 # (Melhoria: Gráfico Adicional - Média de Sono por Dia da Semana)
250 plot_avg_sleep_weekday <- sleep_day %>%
251   mutate(day_of_week = weekdays(sleep_date, abbreviate = FALSE)) %>%
252   group_by(day_of_week) %>%
253   summarise(avg_sleep_minutes = mean(total_sleep_minutes, na.rm = TRUE)) %>%
254   mutate(day_of_week = factor(day_of_week, levels = c("domingo", "segunda-feira", "terça-feira", "quarta-feira", "quinta-feira", "sexta-feira", "sábado")))
255 ggplot(aes(x = day_of_week, y = avg_sleep_minutes, fill = day_of_week)) +
256   geom_bar(stat = "identity") +
257   labs(title = "Média de Minutos de Sono por Dia da Semana",
258        x = "Dia da Semana",
259        y = "Média de Minutos de Sono") +
260   my_theme +
261   theme(legend.position = "none")
262 print(plot_avg_sleep_weekday)
263
264 # (Melhoria: Gráfico Adicional - Distribuição de Minutos Sedentários por Nível de Atividade)
265 plot_sedentary_by_activity <- ggplot(data = final_merged_data_segmented, aes(x = activity_level, y = SedentaryMinutes)) +
266   geom_boxplot() +
267   labs(title = "Minutos Sedentários por Nível de Atividade do Usuário",
268        x = "Nível de Atividade",
269        y = "Minutos Sedentários") +
270   my_theme +
271   theme(legend.position = "none")
272 print(plot_sedentary_by_activity)
```

## 3. Tabela de Funções Comuns (Glossário)

Esta tabela resume as principais funções e operadores utilizados no código R, fornecendo uma referência rápida para sua compreensão.



Atribuição/Função	Descrição
<b>library()</b>	Carrega pacotes (bibliotecas) do R para uso no script.
<b>setwd()</b>	Define o diretório de trabalho atual do R.
<b>dirname()</b>	Retorna o caminho do diretório de um arquivo.
<b>rstudioapi::getActiveDocumentContext()\$path</b>	Retorna o caminho completo do arquivo R atualmente aberto no RStudio.
<b>read_csv()</b>	Lê arquivos CSV, parte do pacote readr (do tidyverse).
<b>%&gt;% (pipe)</b>	Operador que "passa" o resultado da expressão à esquerda como o primeiro argumento da função à direita, tornando o código mais legível.
<b>rename()</b>	Renomeia colunas de um data frame.
<b>mutate()</b>	Adiciona novas colunas ou modifica colunas existentes em um data frame.
<b>mdy() / mdy_hms()</b>	Funções do lubridate para converter strings em formato de data (mês/dia/ano) ou data e hora (mês/dia/ano hora:minuto:segundo).
<b>as.Date()</b>	Converte um objeto para o formato de data.
<b>weekdays()</b>	Extrai o dia da semana de uma data.
<b>as.character()</b>	Converte um objeto para o formato de caractere (texto).
<b>sum(duplicated())</b>	Conta o número de linhas duplicadas em um data frame.
<b>sum(is.na())</b>	Conta o número de valores ausentes (NA) em um data frame.
<b>merge()</b>	Combina dois data frames com base em colunas comuns.
<b>glimpse()</b>	Fornecer uma visão rápida da estrutura de um data frame.
<b>n_distinct()</b>	Conta o número de valores únicos em uma coluna.
<b>summary()</b>	Gera um resumo estatístico para as colunas de um data frame.
<b>mean()</b>	Calcula a média de um vetor numérico.
<b>na.rm = TRUE</b>	Argumento comum em funções estatísticas para remover valores NA antes do cálculo.
<b>cat()</b>	Imprime texto no console.



<code>cor()</code>	Calcula a matriz de correlação entre variáveis numéricas.
<code>drop_na()</code>	Remove linhas que contêm valores NA.
<code>group_by()</code>	Agrupar um data frame por uma ou mais variáveis para realizar operações em grupo.
<code>summarise()</code>	Cria um novo data frame com resumos de dados agrupados.
<code>case_when()</code>	Função condicional útil para criar novas variáveis com base em múltiplas condições.
<code>ggplot()</code>	Inicia a criação de um gráfico no ggplot2.
<code>aes()</code>	Mapeia variáveis do data frame para atributos estéticos do gráfico (e.g., x, y, cor, tamanho).
<code>geom_point()</code>	Adiciona uma camada de pontos (gráfico de dispersão).
<code>geom_smooth()</code>	Adiciona uma linha de suavização (e.g., regressão linear) a um gráfico.
<code>geom_bar()</code>	Adiciona uma camada de barras (gráfico de barras).
<code>geom_boxplot()</code>	Adiciona uma camada de box plots.
<code>labs()</code>	Adiciona ou modifica rótulos e títulos do gráfico.
<code>theme()</code>	Personaliza elementos não-dados do gráfico (e.g., títulos, eixos, legendas).
<code>print()</code>	Exibe um objeto (como um gráfico) no console ou painel de plots.

#### 4. Resultados e Discussão: Insights para a Bellabeat

Com base na análise dos dados, foram identificadas tendências significativas no uso de dispositivos inteligentes que podem ser diretamente aplicadas à estratégia de marketing da **Bellabeat**, com foco no produto **Leaf**.

##### 4.1. Principais Tendências Identificadas

1. Correlação Positiva entre Passos e Calorias Queimadas: Uma relação clara onde o aumento da atividade (passos) leva a um maior gasto calórico. Esta é uma tendência fundamental para motivar usuários a se manterem ativos.

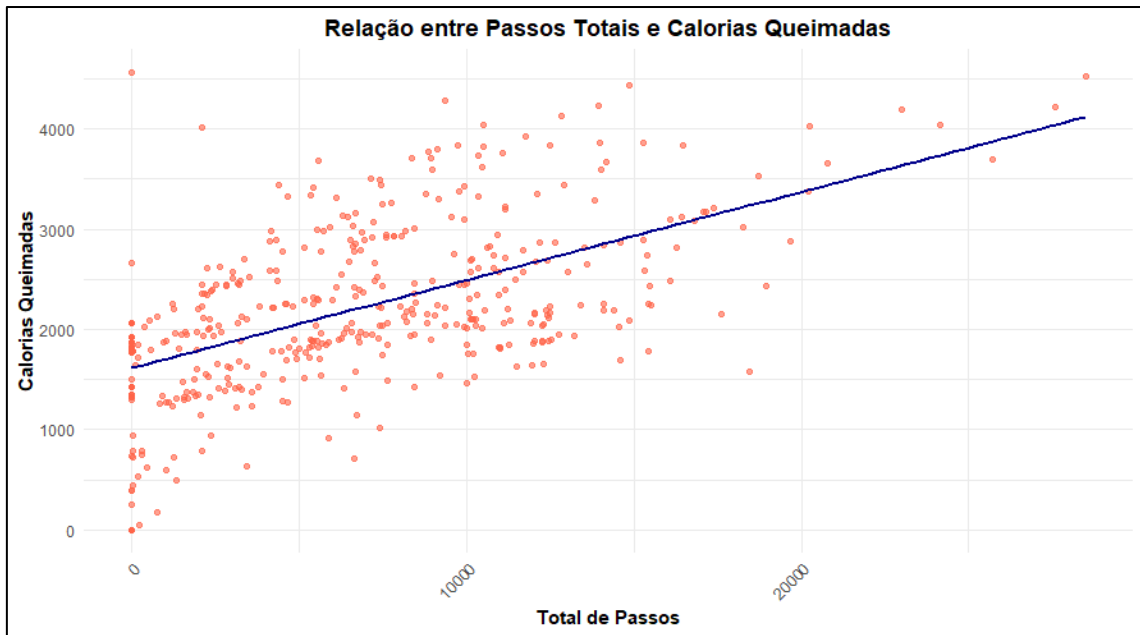


Gráfico 1- Relação entre passos totais e calorias queimadas

2. Padrões de Atividade Variam ao Longo da Semana: A atividade dos usuários não é constante, apresentando flutuações com dias de maior e menor engajamento. Isso sugere oportunidades para intervenções direcionadas.

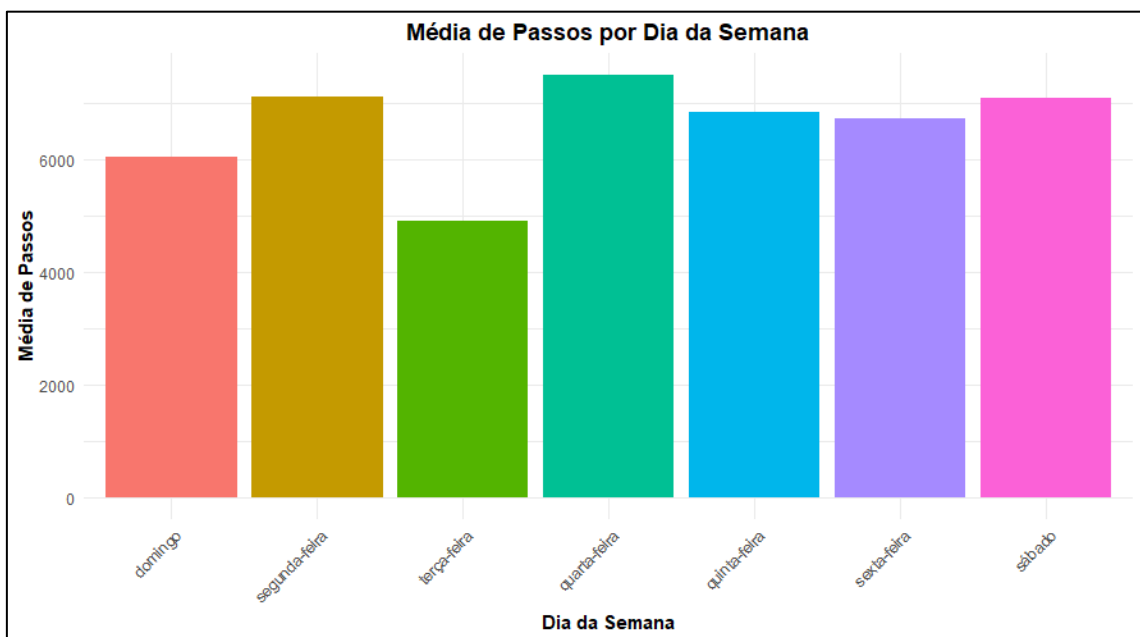


Gráfico 2 - Média de Passos por Dia da Semana

3. Importância do Sono para o Bem-Estar Geral: Os dados de sono revelam padrões diversos, e a qualidade do sono é um fator crítico que pode influenciar os níveis de energia e atividade do dia seguinte.

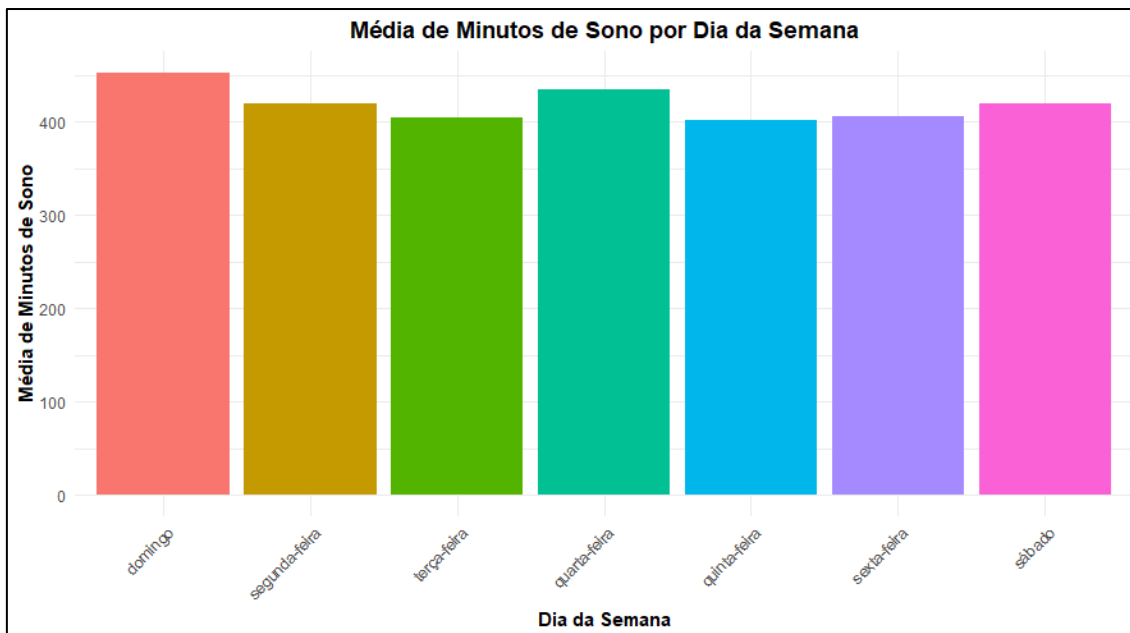


Gráfico 3- Média de Minutos de Sono por Dia da Semana

4.Diferentes Perfis de Usuários: A segmentação dos usuários por nível de atividade (Sedentários, Levemente Ativos, Ativos Moderados, Muito Ativos) demonstra que existem comportamentos e necessidades distintas, exigindo abordagens personalizadas.

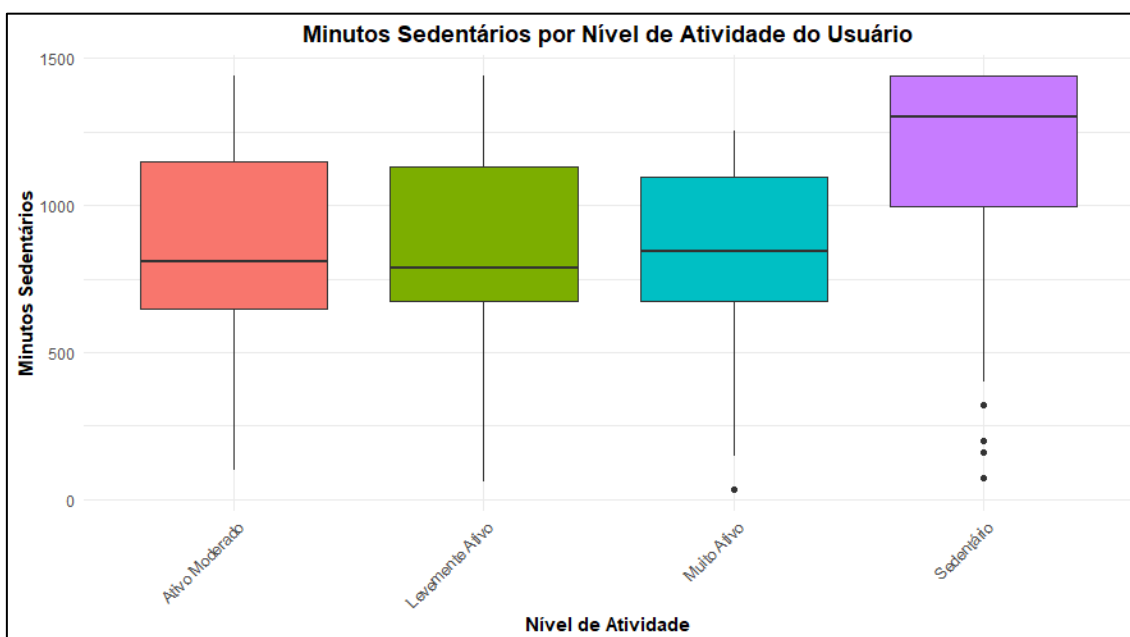


Gráfico 4 - Minutos Sedentários por Nível de Atividade do Usuário

Obs.: Este gráfico ilustra a distribuição dos minutos sedentários diários para cada perfil de usuário. A análise revela que todos os grupos, incluindo os "Muito Ativos", passam uma quantidade significativa de tempo parados (mediana acima de 800 minutos). O grupo "Sedentário" apresenta não só a maior mediana, mas também a maior variabilidade no comportamento. Isso indica que o hábito sedentário é um desafio presente em todos os segmentos de usuários.





## 4.2. Aplicação dos Insights ao Bellabeat Leaf

O Bellabeat Leaf, como um monitor de atividade, sono e estresse, está perfeitamente posicionado para capitalizar sobre essas tendências:

- **Motivação por Resultados:** Usuárias do Leaf podem ser motivadas ao verem a quantificação direta da queima calórica associada aos seus passos, incentivando a manutenção ou aumento da atividade.
- **Incentivo em Dias de Baixa Atividade:** O Leaf pode enviar lembretes e sugestões de atividades leves em dias onde o engajamento tende a ser menor, ajudando as usuárias a manterem a consistência.
- **Otimização do Sono:** Ao monitorar o sono, o Leaf permite que as usuárias compreendam seus padrões e recebam dicas personalizadas para melhorar a qualidade do descanso, impactando positivamente sua energia diária.
- **Experiência Personalizada:** A segmentação de usuários permite que o aplicativo Bellabeat e o Leaf ofereçam recomendações e metas adaptadas ao perfil de atividade de cada usuária, tornando a experiência mais relevante e eficaz.

## 4.3. Recomendações para a Estratégia de Marketing da Bellabeat

1. **Campanhas Focadas em Benefícios Tangíveis:** Destacar como o Leaf quantifica o esforço (passos, calorias) e os benefícios diretos para a saúde e bem-estar. Criar desafios de "passos diários" com recompensas virtuais no aplicativo.
2. **Marketing Contextualizado por Dia da Semana:** Desenvolver campanhas específicas para fins de semana, promovendo atividades leves e relaxantes. Enviar lembretes personalizados para incentivar a atividade em dias de menor engajamento.
3. **Ênfase na Qualidade do Sono:** Campanhas que eduquem sobre a importância do sono e como o Leaf auxilia no monitoramento e melhoria dos hábitos de sono, oferecendo dicas de higiene do sono e meditações guiadas.
4. **Segmentação de Mensagens:** Adaptar a comunicação de marketing aos diferentes perfis de usuários. Para usuárias mais sedentárias, focar nos benefícios da atividade gradual; para as mais ativas, em otimização de desempenho e recuperação.

## 5. Conclusão

Este projeto demonstra o poder da análise de dados de dispositivos inteligentes para gerar insights acionáveis. As tendências identificadas fornecem uma base sólida para a **Bellabeat** refinar sua estratégia de marketing, personalizar a experiência do usuário do **Leaf** e, em última instância, fortalecer sua posição no mercado de tecnologia de bem-estar. A implementação dessas recomendações pode levar a um maior engajamento do usuário e a um crescimento sustentável para a empresa.

Lucas Gonzaga Pereira | [lucasgpec@gmail.com](mailto:lucasgpec@gmail.com)

[Linkedin](#)



[GitHub](#)



[Kaggle](#)

