

Engenharia de Prompt: Modelo comparativo de especificações de celulares © 2024 by Lucas dos Santos Venâncio, Louis Brommelstroet, Vitor Ferraz de Oliveira Martins is licensed under [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)



Membros da equipe:

RA	Nome	Curso de cada membro (SISTEL, Eng. Mecatrônica, etc.)
2128004	Lucas dos Santos Venâncio	BSI
2127997	Louis Brommelstroet	BSI
1426648	Vitor Ferraz de Oliveira Martins	Eng. Mecatrônica

Cliente externo (se houver cliente externo é necessário preencher uma ficha de autorização de imagem)

CPF	Nome
Idade	Observações

Título do projeto:

Modelo comparativo de especificações de celulares

Descrição resumida do projeto (breve contexto, descrição do problema que se deseja resolver, objetivos, metodologia, resultados e conclusão – máximo 500 palavras)

<p>Descrição do problema: Com os inúmeros avanços na tecnologia, uma grande parte da população não possui o conhecimento necessário para efetuar comparações entre quais dispositivos celulares são melhores que outros, conceitos como a velocidade de uma memória RAM, os núcleos de um processador e a quantidade de mAh de uma bateria são apenas o começo quando se fala nessa comparação.</p> <p>Objetivos do projeto: A partir disso a nossa ideia é criarmos um prompt que gere uma explicação breve, ou até mesmo detalhada, entre modelos de celulares, com base em componentes específicos (i.e. tela, bateria, processador, etc).</p> <p>Metodologia: Com base nesse problema teremos que encontrar meios de recuperarmos informações dos componentes que possam ser utilizadas na comparação, além de disponibilizarmos ao usuário a escolha de um período de tempo (desde que respeite o limite do LLM).</p>
--

Resultados:

- Prompt 1: A partir de agora, quando eu disser que necessito das especificações de algum aparelho celular, me refiro a capacidade da bateria, ao modelo do processador com sua quantidade de núcleos e velocidade, a quantidade de memória RAM, ao tamanho da tela e a câmera. Com base nisso, retorne para mim 5 celulares na faixa de R\$1000 a R\$2000, com suas especificações e de acordo com essas especificações qual é o melhor, o melhor celular será aquele com o menor custo possível mas com o melhor processador, bateria e quantidade de memória RAM, a câmera e o tamanho são pontos secundários.

Resultado:

Celulares entre R\$1.000 e R\$2.000: Comparativo e Melhor Escolha

Com base nas suas preferências por processador, bateria e RAM, analisei 5 opções entre R\$1.000 e R\$2.000:

1. POCO X5 Pro 5G:

- **Bateria:** 5000mAh
- **Processador:** Qualcomm Snapdragon 695 5G (8 núcleos, 2.2 GHz)
- **RAM:** 6GB ou 8GB
- **Tela:** 6.67" AMOLED, 120Hz, Full HD+
- **Câmera:** Traseira: 108MP + 8MP + 2MP | Frontal: 16MP
- **Destaques:** Processador potente, bateria grande, tela de alta qualidade.
- **Desvantagem:** Câmera principal não é a melhor da categoria.

2. Redmi Note 12 Pro:

- **Bateria:** 5000mAh
- **Processador:** MediaTek Dimensity 1080 (8 núcleos, 2.6 GHz)
- **RAM:** 6GB, 8GB ou 12GB
- **Tela:** 6.67" AMOLED, 120Hz, Full HD+
- **Câmera:** Traseira: 50MP + 8MP + 2MP | Frontal: 16MP
- **Destaques:** Processador rápido, bastante RAM, bateria grande.
- **Desvantagem:** Tela não tem a melhor resolução.

3. Samsung Galaxy A53 5G:

- **Bateria:** 5000mAh
- **Processador:** Samsung Exynos 1280 (8 núcleos, 2.4 GHz)
- **RAM:** 6GB ou 8GB
- **Tela:** 6.5" Super AMOLED, 120Hz, Full HD+
- **Câmera:** Traseira: 64MP + 12MP + 5MP + 5MP | Frontal: 32MP
- **Destaques:** Câmera versátil com boa qualidade, tela Super AMOLED.
- **Desvantagem:** Processador menos potente que os modelos anteriores.

4. Motorola Edge 30:

- **Bateria:** 4000mAh
- **Processador:** Qualcomm Snapdragon 778G+ 5G (8 núcleos, 2.5 GHz)
- **RAM:** 6GB ou 8GB
- **Tela:** 6.5" pOLED, 144Hz, Full HD+
- **Câmera:** Traseira: 50MP + 13MP | Frontal: 16MP
- **Destaques:** Tela com alta taxa de atualização, processador potente.
- **Desvantagem:** Bateria menor que os outros modelos.

5. realme 9 Pro+:

- **Bateria:** 4500mAh
- **Processador:** MediaTek Dimensity 920 5G (8 núcleos, 2.5 GHz)
- **RAM:** 8GB
- **Tela:** 6.4" Super AMOLED, 90Hz, Full HD+
- **Câmera:** Traseira: 50MP + 8MP + 2MP | Frontal: 16MP
- **Destaques:** Design premium, bom conjunto de câmeras.
- **Desvantagem:** Tela com taxa de atualização menor que alguns modelos.

Conclusão:

O melhor celular entre R\$1.000 e R\$2.000, priorizando processador, bateria e RAM, é o Redmi Note 12 Pro. Ele oferece um processador Dimensity 1080 rápido, 6GB ou 8GB de RAM e bateria de 5000mAh, tudo por um preço acessível.

Vale lembrar que a escolha ideal depende das suas necessidades individuais. Se a câmera for muito importante para você, o Samsung Galaxy A53 5G pode ser uma boa opção. Se você prefere uma tela com alta taxa de atualização, o Motorola Edge 30 é uma boa escolha.

Recomendo que você pesquise mais sobre cada um dos modelos antes de tomar sua decisão final.

Dicas:

- Leia análises e reviews online.
 - Compare preços em diferentes lojas.
 - Verifique se o modelo que você escolher tem as bandas de frequência compatíveis com sua operadora.
- Prompt 2 (Experimental usando llama3 no ollama: <https://ollama.com/library/llama3>) passos:
1. Instale ollama.
 2. Inicie o servidor: `ollama serve`
 3. Rode um modelo (instalação e execução): `ollama run llama3`
 4. Inicie um servidor permitindo acesso CORS: `OLLAMA_ORIGINS=* ollama serve`
 5. Abra o navegador e navegue até uma página de um produto, por exemplo: [buscapé](#)
 6. Pressione F12, execute o código seguinte (código disponível em formato de QR code no apêndice A):

JavaScript

```
// Create a new XMLHttpRequest object
var xhr = new XMLHttpRequest();

// Set the request method and URL
xhr.open('POST', 'http://localhost:11434/v1/chat/completions', true);

// Set the request headers
xhr.setRequestHeader('Content-Type', 'application/json');

// Set the response type
xhr.responseType = 'json';
```

```

// Define what to do when the request is successful
xhr.onload = function() {
  if (xhr.status === 200) {
    var response = xhr.response;
    console.log('Response from ollama:', response);
    console.log(response.choices[0].message.content);
  } else {
    console.error('Error sending data to ollama:', xhr.status);
  }
};

// Define what to do in case of an error
xhr.onerror = function() {
  console.error('Network error while sending data to ollama');
};

// Function to check if the element is visible
function isVisible(element) {
  return element.offsetWidth > 0 && element.offsetHeight > 0;
}

// Function to walk through the DOM and collect text, preserving some layout and
// ignoring non-visible elements
function collectText(node) {
  var text = '';
  if (node.nodeType === Node.TEXT_NODE && isVisible(node.parentElement)) {
    text += node.textContent;
  } else if (node.nodeType === Node.ELEMENT_NODE && isVisible(node)) {
    if (node.tagName !== 'SCRIPT' && node.tagName !== 'STYLE') {
      for (var i = 0; i < node.childNodes.length; i++) {
        text += collectText(node.childNodes[i]);
      }
      if (['P', 'BR', 'HR', 'DIV', 'LI', 'H1', 'H2', 'H3', 'H4', 'H5', 'H6'].indexOf(node.tagName) !== -1) {
        text += '\n'; // Add a newline character after block-level elements
      }
    }
  }
  return text;
}

// Get the text content of the current page while preserving some layout and ignoring
// non-visible elements
var pageText = collectText(document.body).replace(/\s+/g, ' ').trim();

// Get the page's title
var pageTitle = document.title;

// Update the prompt to include the page's title and request for the notebook's
// specifications
var prompt = "You are a web scrapping bot. You can only return JSON code. Extract and
return the product specifications in JSON. \n" +
  "Product: " + pageTitle + "\n";

// Create the data object to send, including the page text in the user's prompt content
var data = {
  "model": "llama3",

```

```

    "messages": [
      //{"role": "system", "content": prompt },
      { "role": "user", "content": pageText + "\n" + prompt },
    ]
  };

  // Convert the data object to a JSON string
  var jsonData = JSON.stringify(data);

  // Send the JSON data to ollama
  xhr.send(jsonData);

```

7. O conteúdo do prompt é: (Texto da página) You are a web scrapping bot. You can only return JSON code. Extract and return the product specifications in JSON.
Product: (Título da página)

8. Obtenha um resultado em JSON como por exemplo:

Here are the product specifications extracted from the text:

```

...
{
  "product_name": "Smartphone Samsung Galaxy A54 5G SM-A546E",
  "ram": 8,
  "storage": 128,
  "camera_type": "Tripla",
  "processor": "Exynos 13.80",
  "processor_speed": {
    "2.0GHz": true,
    "2.4GHz": true
  },
  "battery_capacity": 5000,
  "display_type": "Super AMOLED",
  "display_size": 6.4,
  "display_resolution": "Full HD+",
  "water_resistance": true,
  "weight": 202.0,
  "height": 15.82,
  "width": 7.67
}
...

```

Please note that these specifications are based on the text provided and might not be an exhaustive list of features for this product.

```

>> // Create a new XMLHttpRequest object
var xhr = new XMLHttpRequest();

// Set the request method and URL
xhr.open('POST', 'http://localhost:11434/v1/chat/completions', true);-
<- undefined

Response from ollama: ▶ Object { id: "chatcmpl-861", object: "chat.completion", created: 1714605186, model: "llama3", system_fingerprint: "fp_ollama", choices: (1) [...], u
Here are the product specifications extracted from the text:

...
{
  "product_name": "Smartphone Samsung Galaxy A54 5G SM-A546E",
  "ram": 8,
  "storage": 128,
  "camera_type": "Tripla",
  "processor": "Exynos 13.80",
  "processor_speed": {
    "2.0GHz": true,
    "2.4GHz": true
  },
  "battery_capacity": 5000,
  "display_type": "Super AMOLED",
  "display_size": 6.4,
  "display_resolution": "Full HD+",
  "water_resistance": true,
  "weight": 202.0,
  "height": 15.82,
  "width": 7.67
}
...

Please note that these specifications are based on the text provided and might not be an exhaustive list of features for this product.
>>

```

Informações do modelo:

llama3:latest a6990ed6be41 4.7 GB

arquitetura: llama

parametros: 8B (bilhões)

quantização: 4 bits

tamanho: 4.7GB

Comparação com Gemini Pro 1.5: Modelos locais como o llama3 podem ser interessantes porque podem trazer resultados sem um limite ou custo de utilização de uma API, não dependendo da disponibilidade da API, apesar de exigir recursos computacionais como preferencialmente uma GPU que possa carregar o modelo em sua memória. Como estes modelos não são tão poderosos como modelos muito maiores como o Gemini Pro 1.5, existem algumas limitações como um contexto de 2048 tokens, respostas que têm uma variabilidade maior nas respostas aceitáveis, e uma probabilidade maior de errar ou acertar inconsistentemente, o que exige um maior trabalho ao criar o prompt. Uma das estratégias aqui utilizadas foi mover as instruções para o final, o que possibilitou acertar todas as vezes testadas com o site acima. O prompt acima utilizou 1605 tokens. Para comparação, o modelo Gemini Pro 1.5 possui um contexto de 1,048,576 tokens, o que permitiu realizar o prompt abaixo.

- Prompt 3:

Return specifications from the relevant product in JSON format: (código HTML)

Resultado:

```

{
  "Marca": "Samsung",
  "Modelo": "SM-A546E 8GB RAM",
  "Linha": "Galaxy",
  "Série": "A54 5G",

```

```

"Tipo de Aparelho": "Smartphone",
"Quantidade de Chip": "2 Chips",
"Câmera Frontal": "Sim",
"Resolução da Câmera Frontal": "32.0 MP",
"Abertura da Câmera Frontal": "f/2.2",
"Resolução da Câmera Traseira": "50.0 + 12.0 + 5.0 MP",
"Abertura da Câmera Traseira": "f/1.8 + f/2.2 + f/2.4 (macro)",
"Filma em(Câmera Frontal)": "4K",
"Filma em(Câmera Traseira)": "4K",
"Diferenciais da Câmera": ["Autofoco", "Câmera Tripla", "Flash", "HDR",
"Slow Motion"],
"Sistema Operacional": "Android 13",
"Processador": "Exynos 1380",
"Núcleo de Processador": "Octa-Core",
"Velocidade do Processador": ["2.0GHz", "2.4GHz"],
"Memória Ram": "8.0GB",
"Memória Interna": "128GB",
"Suporta Cartão microSD de até": "1024.0GB",
"Capacidade da Bateria": "5000mAh",
"Tipo de Tela": "Super AMOLED",
"Tamanho da Tela": "6.4\"",
"Resolução da Tela": "Full HD+",
"Taxa de Atualização": "120.0Hz",
"Acesso à Internet": ["4G", "5G", "Wi-Fi (2.4 GHz e 5.0 GHz)", "Wi-Fi
(6.0 GHz)"],
"Conectividade": ["Bluetooth", "NFC", "USB Type-C"],
"Recursos e Funções": ["Acelerômetro", "Giroscópio", "Leitor Biométrico
na Tela", "Resistente à água IP67", "Sensor de Proximidade", "Som
Estéreo"],
"Diferenciais desse Produto": ["Bandeja Híbrida"],
"Altura": "15.82cm",
"Largura": "7.67cm",
"Profundidade": "0.82cm",
"Peso": "202.0g"
}

```

Passos:

1. Visite uma página de um produto, por exemplo:
<https://www.buscape.com.br/celular/smartphone-samsung-galaxy-a54-5g-sm-a54-6e-8-gb-128gb-camera-tripla-exynos-1380-2-chips-android-13>
2. Vamos copiar o código HTML, pressione F12 e execute o comando:
`copy(document.documentElement.outerHTML);`
ou copie o código HTML manualmente utilizando as ferramentas de desenvolvedor.

3. Crie uma conversa nova no [gemini](#), use o seguinte prompt:
Return specifications from the relevant product in JSON format: (Pressione CTRL+V)
 4. Obtenha um resultado como o exemplo acima.
- Prompt 4 (Combinando informações obtidas do prompt 3 (ou 2) e utilizando o prompt 1) passos:

1. Adapte o prompt 3 para incluir o preço:

Return specifications from the relevant product in JSON format, including price:
(HTML)

2. Execute o prompt 3 para cada produto que desejar, junte todos os resultados JSON de todos os produtos que serão analisados em único texto.
3. Adapte o prompt 1 para analisar os produtos:

A partir de agora, quando eu disser que necessito das especificações de algum aparelho celular, me refiro a capacidade da bateria, ao modelo do processador com sua quantidade de núcleos e velocidade, a quantidade de memória RAM, ao tamanho da tela e a câmera. Com base nisso, compare os celulares abaixo, com suas especificações e de acordo com essas especificações qual é o melhor, o melhor celular será aquele com o menor custo possível mas com o melhor processador, bateria e quantidade de memória RAM, a câmera e o tamanho são pontos secundários.

(Texto juntando cada resultado JSON de cada produto)

4. Resultado:

Comparação de Celulares:

Com base nas especificações fornecidas e priorizando processador, bateria e RAM, seguem as análises:

Samsung Galaxy A54 5G:

- **Pontos Positivos:** Boa bateria (5000mAh), 8GB de RAM e processador Octa-Core de até 2.4GHz. Câmera tripla e tela Super AMOLED de 6.4".
- **Pontos Negativos:** Processador Exynos 1380 pode ser inferior em desempenho aos concorrentes.

Motorola Moto G84:

- **Pontos Positivos:** Boa bateria (5000mAh), 8GB de RAM, processador Snapdragon 695 (considerado superior ao Exynos 1380) e preço mais baixo. Tela POLED de 6.5".
- **Pontos Negativos:** Câmera dupla pode ser inferior em versatilidade.

Apple iPhone 13:

- **Pontos Negativos:** Menor quantidade de RAM (4GB), bateria consideravelmente menor (3240mAh) e preço significativamente mais alto.
- **Pontos Positivos:** Processador A15 Bionic é o mais potente entre os três, tela Super Retina XDR de 6.1" com resolução 2K e câmera dupla com recursos avançados.

Conclusão:

Considerando o critério de melhor custo-benefício com foco em processador, bateria e RAM, o **Motorola Moto G84** parece ser a melhor opção. Ele oferece um bom equilíbrio entre desempenho, recursos e preço.

Observações:

- A escolha final pode depender de suas necessidades individuais e preferências de marca.
- É recomendável pesquisar reviews e comparações mais detalhadas antes de tomar uma decisão.

Conclusão:

Resumo de cada reunião da equipe (incluir uma linha para cada dia que a equipe se reunir)

Reunião 1 (18/04/2024)

Definição de projeto e início dos prompts experimentais.

Reunião 2 (25/04/2024)

Estruturação do prompt e análise de obtenção e repetibilidade de resultados.

Imagem de uma reunião da equipe (data) (se houver cliente externo, incluir imagem com o cliente externo)



Reunião dia 18/04/2024

(imagem em sala)

Reunião dia 25/04/2024

Imagem do dia apresentação do projeto em sala de aula

Ainda:

1. É necessário incluir cópia pdf do prompt desenvolvido
2. Se houver cliente externo, incluir ficha de avaliação do cliente externo

3. Incluir pdf do ppt da apresentação final

Apêndice A: QR Code contendo o código do Prompt 2

