Blueprint do Projeto: Compressão de Pesos de LLM (v2)

Fase 1: Análise Exploratória e Caracterização da Fonte (Concluída)

- **1.1 Engenharia Reversa:**
  - ☒ Identificar estrutura: Header JSON + Payload Binário contíguo.
  - ☒ Identificar tipo de dados: `BF16` (2 bytes: 1 sinal+expoente, 1 mantissa).
- **1.2 Análise Estatística:**
  - ☒ Calcular Entropia Global: ~6.22 bits/byte.
  - ☒ Calcular Entropia Condicional: ~5.36 bits/byte.
  - ☒ **Byte-Splitting (Descoberta Chave):**
    - Entropia MSB (Expoente): **2.71 bits/byte** (Alta redundância).
    - Entropia LSB (Mantissa): **7.96 bits/byte** (Ruído quase aleatório).

Fase 2: Benchmarking de Referência (Em Progresso)

Objetivo: Estabelecer as métricas a bater.

- **Objetivo 2.1: Testes com Compressores Padrão:**
  - Executar `gzip` (níveis -1 e -9), `bzip2`, `xz` e `zstd` sobre o ficheiro original.
  - Registar: Tamanho Final, Tempo de Compressão/Descompressão, Pico de RAM.
- **Objetivo 2.2: Testes Específicos (Opcional):**
  - Testar ferramentas especializadas em floats (`fpzip` ou `zfp`) apenas para comparação teórica, se houver tempo.

Fase 3: Desenvolvimento do Codec "Split-Stream" (O Core)

Estratégia Definida: Arquitetura Híbrida baseada na separação MSB/LSB.

- **Objetivo 3.1: Módulo de Pré-processamento (Splitter):**
  - Criar um programa que leia o ficheiro original em blocos (ex: 1MB) e separe os dados em dois buffers/ficheiros temporários: `stream_msb` e `stream_lsb`.
  - Garantir eficiência de memória (não carregar 1GB de uma vez).
- **Objetivo 3.2: Compressão do Canal MSB (Alta Compressão):**
  - **Passo A (Transformação):** Implementar **Predição Delta** ($r_n = x_n - x_{n-1}$) nos bytes MSB.
  - **Passo B (Verificação):** Medir a entropia dos resíduos gerados. Meta: < 2.0 bits/byte.

  - **Passo C (Codificação):** Implementar ou integrar um codificador de entropia (Huffman ou Aritmético) para os resíduos.
- **Objetivo 3.3: Compressão do Canal LSB (Baixa Complexidade):**
  - Devido à alta entropia (~7.96), testar duas abordagens:
    1. *Raw Storage:* Guardar sem compressão (custo computacional zero).
    2. *Lightweight:* Usar um algoritmo muito rápido (ex: RLE ou LZ4) apenas para apanhar eventuais sequências de zeros.
- **Objetivo 3.4: Empacotamento (Bitstream Final):**
  - Definir o formato do ficheiro comprimido `.sc` (Ex: `[Header Tamanho][Bloco Comprimido MSB][Bloco LSB]...`).

Fase 4: Otimização e Pontos de Operação

Objetivo: Criar as variantes "Fast" vs "Best" exigidas no enunciado.

- **Objetivo 4.1: Tuning de Parâmetros:**
  - **Modo "Fast":** Split + Delta (MSB) + Huffman Estático + LSB Raw.
  - **Modo "Best":** Split + Delta (MSB) + Aritmética Adaptativa + LSB (tentativa LZ).
- **Objetivo 4.2: Gestão de Memória:**
  - Refinar o tamanho dos *chunks* de leitura para garantir que o compressor funciona em máquinas com pouca RAM.

Fase 5: Relatório e Apresentação

- **Objetivo 5.1: Escrita Técnica:**
  - Documentar a implementação do *Splitter* e do *Preditor*.
  - Comparar os resultados finais com o Benchmark da Fase 2.
- **Objetivo 5.2: Apresentação:**
  - Preparar slides focados na decisão de arquitetura ("Porquê separar os bytes?").