Claro! Aqui está um **artigo técnico** detalhado com diretrizes para perfurar cartões de 80 colunas com o objetivo de criar um programa de impressão de relatório de funcionários da folha de pagamento.

**Diretrizes Técnicas para Perfuração de Cartões de 80 Colunas: Programa de Impressão de Relatório de Funcionários da Folha de Pagamento**

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

**1. Introdução**

Os cartões perfurados de 80 colunas foram amplamente utilizados em sistemas de computação antigos, especialmente em mainframes IBM como os da série 360/370. Embora obsoletos na prática moderna, compreender sua estrutura e uso ainda é relevante em ambientes de simulação, restauração de sistemas legados ou ensino de fundamentos da computação. Este artigo descreve as diretrizes para a criação de um programa de impressão de relatório de funcionários utilizando cartões de 80 colunas.

**2. Estrutura do Cartão de 80 Colunas**

Cada cartão contém 80 colunas numeradas de 1 a 80, e cada coluna pode representar um caractere alfanumérico. Os cartões são organizados em sequências, onde cada linha de código-fonte ou dado ocupa um cartão.

**3. Requisitos do Programa**

**Objetivo:** Gerar um relatório impresso listando os funcionários de uma folha de pagamento.

**Dados esperados por funcionário:**

* Código do Funcionário (6 dígitos)
* Nome (20 caracteres)
* Cargo (15 caracteres)
* Departamento (10 caracteres)
* Salário (9 dígitos com 2 casas decimais)
* Data de Admissão (AAAAMMDD - 8 dígitos)

**Total: 68 caracteres por registro.**

**4. Diretrizes para Perfuração**

**4.1. Formato dos Cartões**

* Use cartões em branco de 80 colunas.
* Utilize uma perfuradora de cartões padrão (como IBM 029).
* Preencha com espaços as colunas não utilizadas (colunas 69 a 80).

**4.2. Cartões de Código-Fonte**

O programa pode ser escrito em COBOL ou FORTRAN, com uma estrutura padrão:

**Exemplo (COBOL):**

css

CopiarEditar

000100 IDENTIFICATION DIVISION.

000200 PROGRAM-ID. PAYROLL-REPORT.

000300 ENVIRONMENT DIVISION.

000400 INPUT-OUTPUT SECTION.

000500 FILE-CONTROL.

000600 SELECT EMPLOYEE-FILE ASSIGN TO CARD-READER.

000700 SELECT REPORT-FILE ASSIGN TO PRINTER.

000800 DATA DIVISION.

...

**Observações:**

* Colunas 1-6: Número de sequência (opcional, mas recomendável).
* Coluna 7: Indicação de comentário (\*) ou continuação.
* Colunas 8-72: Código-fonte.
* Colunas 73-80: Número de sequência de controle (opcional).

**4.3. Cartões de Dados**

Para cada funcionário, crie um cartão de dados com a seguinte disposição:

| **Campo** | **Início** | **Fim** | **Formato** |
| --- | --- | --- | --- |
| Código Funcionário | 1 | 6 | Númerico, zeros à esquerda |
| Nome | 7 | 26 | Alfabético, alinhado à esquerda |
| Cargo | 27 | 41 | Alfabético |
| Departamento | 42 | 51 | Alfabético |
| Salário | 52 | 60 | Númerico sem separador |
| Data de Admissão | 61 | 68 | AAAAMMDD |
| Espaço | 69 | 80 | Preenchido com espaços |

**Exemplo de conteúdo:**

CopiarEditar

000123JOAO DA SILVA ANALISTA TI 00045000020200115

**5. Impressão do Relatório**

O programa deve ler os cartões de dados, organizar os registros e imprimir o relatório em formato tabular. A saída pode ser impressa em uma impressora de linha ou enviada para arquivo.

**Exemplo de saída:**

markdown

CopiarEditar

---------------------------------------------------------------------

CÓDIGO NOME CARGO DEPTO SALÁRIO ADMISSÃO

---------------------------------------------------------------------

000123 JOAO DA SILVA ANALISTA TI 4500.00 2020-01-15

000124 MARIA FONSECA GERENTE RH 7250.00 2018-03-22

...

**6. Boas Práticas**

* Utilize uma planilha ou editor com marcação de colunas para validar o conteúdo antes da perfuração.
* Sempre mantenha uma cópia de segurança dos cartões (em imagem ou fita magnética, se possível).
* Use validação cruzada ao carregar os cartões no sistema para evitar erros de perfuração.

**7. Considerações Finais**

Embora os cartões perfurados sejam obsoletos, ainda têm valor didático e histórico. Este guia fornece um exemplo prático e funcional de como usá-los para criar um sistema simples de relatório de folha de pagamento. Ele pode ser adaptado para simulações com ferramentas como o Hercules (emulador IBM mainframe) e linguagens como COBOL ou FORTRAN.

Se você quiser que eu gere um exemplo completo em COBOL ou FORTRAN para esse caso, posso fazer isso também! Deseja seguir por esse caminho?