

Dia 07/04

NadsonJR: Você é um desenvolvedor senior de Cobol com conhecimento em COBOL VSAM e DB2 Oracle, que acabou de entrar em um novo projeto, agora eu preciso que realize uma análise de um código

para contextualizar o projeto

GitHub Copilot: Claro! Vou realizar uma análise inicial do código COBOL fornecido para contextualizar o projeto. Aqui está um resumo do que o programa faz, com base no código:

- --

**Objetivo do Programa**

O programa `CIPB106` foi desenvolvido para gerar estatísticas de utilização por filial, comparando dados entre os sistemas CIP e CIC. Ele processa registros de entrada, realiza cálculos e gera um relatório detalhado com informações de atualizações e estatísticas.

- --

**Estrutura Geral**

1. **Divisão de Identificação:**

- O programa foi escrito em 2002 e sofreu diversas manutenções ao longo do tempo, como inclusão de novas colunas no relatório e tratamento de novos cenários.

2. **Divisão de Ambiente:**

- Define arquivos de entrada e saída:
- `CICA600`: Arquivo de entrada sequencial.
- `TAB539`: Arquivo indexado para consultas.
- `CIPR1061`: Arquivo de saída para o relatório.
- Utiliza `SPECIAL-NAMES` para configurar o ponto decimal como vírgula.

3. ****Divisão de Dados****:

- ****Seção de Arquivos****:

- Define os layouts dos arquivos mencionados.

- ****Seção de Trabalho****:

- Contém variáveis auxiliares, contadores, tabelas e estruturas para geração do relatório.

- Inclui tabelas para armazenar estatísticas por filial (`TAB-FILIAL`).

- ****Books e Copys****:

- Utiliza `COPY` para incluir definições externas, como `CICB600` e `CIPC109`.

4. ****Divisão de Procedimentos****:

- ****Fluxo Principal****:

- Realiza as seguintes etapas:

1. ****Inicialização**** (`100-INICIO`): Abre arquivos, inicializa variáveis e configura data/hora.

2. ****Leitura de Registros**** (`200-LER-CICA600`): Lê registros do arquivo `CICA600` e realiza validações.

3. ****Processamento**** (`500-PROCESSAR`): Processa os registros lidos, atualiza contadores e consulta o arquivo indexado `TAB539` para validações adicionais.

4. ****Descarga de Dados**** (`600-DESCARGA`): Gera o relatório com base nos dados processados.

5. ****Finalização**** (`900-TERMINO`): Fecha arquivos e imprime totais.

- ****Tratamento de Erros****:

- Seções como `999-ABEND` lidam com erros, exibindo mensagens e

encerrando o programa.

- --

**Detalhes Importantes**

1. **Arquivos e Status:**

- O programa verifica o status (`FILE STATUS`) após operações de abertura, leitura e fechamento de arquivos. Erros são tratados com mensagens específicas e chamadas para rotinas deabend.

2. **Relatório Gerado:**

- O relatório inclui:
- Estatísticas de atualizações por filial (CIC e CIP).
- Totais gerais e desconsiderações (registros inválidos ou rejeitados).
- Mensagens de erro, se aplicável.

3. **Consultas ao Arquivo Indexado:**

- O arquivo `TABA539` é consultado para validar usuários e filiais. Caso o registro não seja encontrado, o programa registra o erro e continua o processamento.

4. **Manutenções e Expansões:**

- O código possui diversas alterações documentadas, como:
- Inclusão de colunas no relatório (`CIP-TELEMARKETING`, `CIP-AUTOMATICO`).
- Tratamento de novos cenários, como usuários não encontrados.

5. **Chamada de Serviço Externo:**

- A seção `520-TRATA-CIP` realiza uma chamada a um serviço externo (`CIPS109`) para tratar registros do tipo `CIP`.

- --

**Contexto do Projeto**

Com base no código, o projeto parece estar relacionado à análise e controle de atualizações realizadas em sistemas de gestão de clientes (CIP e CIC). O programa processa grandes volumes de dados, valida informações e gera relatórios detalhados para acompanhamento e auditoria.

- --

**Próximos Passos**

1. **Entendimento do Fluxo de Dados:**

- Verificar os layouts dos arquivos (`CICA600`, `TABA539`) e entender como os dados são alimentados.
- Confirmar o formato e destino do relatório gerado.

2. **Validação de Requisitos:**

- Revisar as manutenções documentadas para garantir que o programa atende aos requisitos atuais.

3. **Testes e Depuração:**

- Executar o programa com dados reais ou simulados para validar o comportamento e identificar possíveis melhorias.

4. **Documentação e Atualizações:**

- Atualizar a documentação do programa, se necessário, para refletir as alterações realizadas.

Se precisar de uma análise mais detalhada de alguma parte específica do código ou de ajuda para realizar alterações, é só avisar!

NadsonJR: Baseado na analise anterior

1- Realize um code-review entre [CIPB106 - F1rst.cbl] e [cipb106 - Eccox.CBL]

2- Encontre erros nas linhas onde encontra-se a marcação da Eccox no arquivo [cipb106 - Eccox.CBL] e me indique linha a linha onde acontece

2.1 Preciso que procure por coisas como redundancia ou erro de lógica

3- Verifique se o código implementado pela Eccox tem coerencia com a linguagem COBOL/VSAM/Db2 Oracle.

4- Não quero boas praticas de código, estou procurando por erros que afetam diretamente o funcionamento da aplicação

GitHub Copilot: **### **Code Review entre os arquivos [CIPB106 - F1rst.CBL] e [cipb106 - Eccox.CBL]****

**1. Diferenças Gerais**

- O arquivo `cipb106 - Eccox.CBL` contém marcações específicas da Eccox (`Eccox *`) indicando alterações ou comentários adicionais.
- O arquivo `CIPB106 - F1rst.CBL` parece ser a versão original ou uma versão anterior do programa.
- A Eccox adicionou lógica para manipulação de SQL (`EXEC SQL`) e tratamento de `SQLCODE`, enquanto o arquivo original utiliza apenas operações de leitura VSAM.

- --

**2. Erros e Redundâncias no arquivo [cipb106 - Eccox.CBL]**

**Linha 00460000 a 00510000**

```cobol

Eccox \*   SELECT TABA539

Eccox \*   ASSIGN TO TABA539

Eccox \* ORGANIZATION IS INDEXED (pode ser sequencial tbm)  
Eccox \* ACCESS IS RANDOM  
Eccox \* RECORD KEY IS ST-CHV-ARQ-539  
Eccox \* FILE STATUS FS-TABA539.

...

- **\*\*Erro:\*\*** Comentário indica que o arquivo `TABA539` pode ser sequencial, mas a definição de `ORGANIZATION IS INDEXED` é incompatível com acesso sequencial. Isso pode causar erro de compilação ou execução.
- **\*\*Impacto:\*\*** Afeta diretamente o funcionamento se o arquivo for tratado como sequencial em algum momento.

- --

#### #### **\*\*Linha 03310000 a 03370000\*\***

```cobol

Eccox * OPEN INPUT CICA600 TABA539
Eccox * OUTPUT CIPR1061.
Eccox OPEN INPUT CICA600
Eccox OUTPUT CIPR1061.
Eccox MOVE ZEROS TO SQLCODE.
Eccox PERFORM ECX-SQLCODE-PARA-FSTATUS.
Eccox MOVE ECX-WS-FSTATUS TO FS-TABA539.

...

- ****Erro:**** A lógica de `MOVE ZEROS TO SQLCODE` e `PERFORM ECX-SQLCODE-PARA-FSTATUS` é redundante, pois o arquivo `TABA539` não foi aberto. `FS-TABA539` não será atualizado corretamente.

- ****Impacto:**** Pode causar inconsistência no status do arquivo
`TABA539`.

- - -

**Linha 04550000 a 04850000**

```cobol

Eccox \* READ TABA539.

Eccox MOVE DCLTAB217-TABA539 TO RG-TABA539-539.

Eccox SET ECX-TIPO-OUTROS TO TRUE.

Eccox EXEC SQL

Eccox SELECT

Eccox ST\_CHV\_ARQ\_539,

Eccox CD\_PASS\_USUAR\_539,

Eccox NM\_USUAR\_539,

Eccox NM\_DEPTO\_USUAR\_539,

Eccox NO\_RAMAL\_USUAR\_539,

Eccox CD\_EMP\_539,

Eccox CD\_FILIAL\_TAB\_539,

Eccox CD\_LOCAL\_USUAR\_539,

Eccox NM\_FUNC\_USU\_539

Eccox INTO

Eccox :TAB217-ST-CHV-ARQ-539,

Eccox :TAB217-CD-PASS-USUAR-539,

Eccox :TAB217-NM-USUAR-539,

Eccox :TAB217-NM-DEPTO-USUAR-539,

```

Eccox :TAB217-NO-RAMAL-USUAR-539,
Eccox :TAB217-CD-EMP-539,
Eccox :TAB217-CD-FILIAL-TAB-539,
Eccox :TAB217-CD-LOCAL-USUAR-539,
Eccox :TAB217-NM-FUNC-USU-539
Eccox FROM TAB217_TABA539
Eccox WHERE SG_USUAR_539 = :TAB217-ST-CHV-ARQ-539
Eccox END-EXEC.
Eccox PERFORM ECX-SQLCODE-PARA-FSTATUS.
Eccox MOVE ECX-WS-FSTATUS TO FS-TABA539.
Eccox MOVE DCLTAB217-TABA539 TO RG-TABA539-539.
...

```

- **\*\*Erro 1:\*\*** A instrução `READ TABA539` foi substituída por `EXEC SQL SELECT`, mas a lógica de manipulação de `FS-TABA539` ainda está presente. Isso é incoerente, pois `FS-TABA539` não é atualizado por operações SQL.
- **\*\*Erro 2:\*\*** A cláusula `WHERE SG\_USUAR\_539 = :TAB217-ST-CHV-ARQ-539` pode causar erro se `SG\_USUAR\_539` não for inicializado corretamente.
- **\*\*Impacto:\*\*** Pode gerar falhas de execução ou resultados incorretos.
- --

**#### \*\*Linha 07060000 a 07100000\*\***

```

```cobol

```

```

Eccox *    CLOSE CICA600 TABA539.
Eccox      CLOSE    CICA600.

```


Eccox MOVE ZEROS TO SQLCODE.

Eccox PERFORM ECX-SQLCODE-PARA-FSTATUS.

Eccox MOVE ECX-WS-FSTATUS TO FS-TABA539.

...

- ****Erro:**** O arquivo `TABA539` não é fechado explicitamente, mas o status `FS-TABA539` é manipulado. Isso pode causar erro de lógica ou inconsistência no fechamento do arquivo.
- ****Impacto:**** Pode deixar o arquivo `TABA539` aberto, causando problemas em execuções subsequentes.
- --

**3. Coerência com COBOL/VSAM/DB2 Oracle**

- ****SQLCODE e File-Status:**** A Eccox tenta mapear `SQLCODE` para `FS-TABA539` usando a rotina `ECX-SQLCODE-PARA-FSTATUS`. Isso não é coerente, pois `SQLCODE` é específico para operações SQL e não deve ser usado para arquivos VSAM.
- ****Manipulação de Arquivos:**** A lógica de abertura e fechamento de arquivos (`OPEN` e `CLOSE`) está parcialmente implementada. O arquivo `TABA539` não é tratado adequadamente.
- ****Execução de SQL:**** A instrução `EXEC SQL` está correta em termos de sintaxe, mas a lógica de manipulação de resultados (`PERFORM ECX-SQLCODE-PARA-FSTATUS`) é redundante e pode causar inconsistências.
- --

**Resumo dos Problemas Identificados**

1. ****Redundância:**** Manipulação de `SQLCODE` e `FS-TABA539` simultaneamente.
2. ****Erro de Lógica:**** Arquivo `TABA539` não é tratado corretamente

(abertura, leitura e fechamento).

3. ****Incoerência:**** Uso de `SQLCODE` para atualizar `FS-TABA539` não é compatível com COBOL/VSAM.

4. ****Possível Falha SQL:**** Falta de inicialização de variáveis usadas em cláusulas `WHERE`.

Se precisar de mais detalhes ou ajustes, é só avisar!

NadsonJR: Preciso que refaça a análise, e ignore codigos comentados, levando somente os que realmente estão atuando na aplicação, outro ponto é, para cada erro encontrado promova uma solução valida pensando em longevidadea

GitHub Copilot: **### **Revisão Atualizada do Arquivo [cipb106 - Eccox.CBL]****

Agora, considerando apenas o código ativo (não comentado) no arquivo [cipb106 - Eccox.CBL], realizarei a análise para identificar erros que afetam diretamente o funcionamento da aplicação. Para cada erro encontrado, apresentarei uma solução válida, considerando a longevidade e a coerência com COBOL/VSAM/DB2 Oracle.

- --

**1. Análise de Erros no Código Ativo**

**1.1 Linha 03330000 a 03370000**

```cobol

OPEN INPUT CICA600

OUTPUT CIPR1061.

MOVE ZEROS TO SQLCODE.

PERFORM ECX-SQLCODE-PARA-FSTATUS.

MOVE ECX-WS-FSTATUS TO FS-TABA539.

...

- **\*\*Erro Identificado:\*\***

- O arquivo `TABA539` não é aberto explicitamente, mas o `FS-TABA539` é manipulado. Isso pode causar inconsistências, pois `FS-TABA539` não será atualizado corretamente sem a abertura do arquivo.

- **\*\*Solução Proposta:\*\***

- Adicionar a abertura explícita do arquivo `TABA539` antes de manipular seu `FILE STATUS`. Isso garante que o arquivo esteja disponível para operações subsequentes.

```cobol

OPEN INPUT CICA600

INPUT TABA539

OUTPUT CIPR1061.

MOVE ZEROS TO SQLCODE.

PERFORM ECX-SQLCODE-PARA-FSTATUS.

MOVE ECX-WS-FSTATUS TO FS-TABA539.

...

- --

**1.2 Linha 04560000 a 04850000**

```cobol

MOVE DCLTAB217-TABA539 TO RG-TABA539-539.

SET ECX-TIPO-OUTROS TO TRUE.

EXEC SQL

```
SELECT
ST_CHV_ARQ_539,
CD_PASS_USUAR_539,
NM_USUAR_539,
NM_DEPTO_USUAR_539,
NO_RAMAL_USUAR_539,
CD_EMP_539,
CD_FILIAL_TAB_539,
CD_LOCAL_USUAR_539,
NM_FUNC_USU_539
INTO
:TAB217-ST-CHV-ARQ-539,
:TAB217-CD-PASS-USUAR-539,
:TAB217-NM-USUAR-539,
:TAB217-NM-DEPTO-USUAR-539,
:TAB217-NO-RAMAL-USUAR-539,
:TAB217-CD-EMP-539,
:TAB217-CD-FILIAL-TAB-539,
:TAB217-CD-LOCAL-USUAR-539,
:TAB217-NM-FUNC-USU-539
FROM TAB217_TABA539
WHERE SG_USUAR_539 = :TAB217-ST-CHV-ARQ-539
END-EXEC.

PERFORM ECX-SQLCODE-PARA-FSTATUS.
```

MOVE ECX-WS-FSTATUS TO FS-TABA539.

MOVE DCLTAB217-TABA539 TO RG-TABA539-539.

...

- **\*\*Erro Identificado:\*\***

- A manipulação de `FS-TABA539` após a execução de SQL (`EXEC SQL`) é redundante e incorreta. `FS-TABA539` é usado para operações VSAM, enquanto `SQLCODE` é específico para SQL.

- A cláusula `WHERE SG\_USUAR\_539 = :TAB217-ST-CHV-ARQ-539` pode falhar se `SG\_USUAR\_539` não for inicializado corretamente.

- **\*\*Solução Proposta:\*\***

- Remover a manipulação de `FS-TABA539` após a execução de SQL e tratar apenas o `SQLCODE`.

- Garantir que `SG\_USUAR\_539` esteja inicializado antes da execução do SQL.

```cobol

MOVE DCLTAB217-TABA539 TO RG-TABA539-539.

SET ECX-TIPO-OUTROS TO TRUE.

IF SG_USUAR_539 NOT EQUAL SPACES

EXEC SQL

SELECT

ST_CHV_ARQ_539,

CD_PASS_USUAR_539,

NM_USUAR_539,

NM_DEPTO_USUAR_539,

NO_RAMAL_USUAR_539,

```
CD_EMP_539,
CD_FILIAL_TAB_539,
CD_LOCAL_USUAR_539,
NM_FUNC_USU_539
INTO
:TAB217-ST-CHV-ARQ-539,
:TAB217-CD-PASS-USUAR-539,
:TAB217-NM-USUAR-539,
:TAB217-NM-DEPTO-USUAR-539,
:TAB217-NO-RAMAL-USUAR-539,
:TAB217-CD-EMP-539,
:TAB217-CD-FILIAL-TAB-539,
:TAB217-CD-LOCAL-USUAR-539,
:TAB217-NM-FUNC-USU-539
FROM TAB217_TABA539
WHERE SG_USUAR_539 = :TAB217-ST-CHV-ARQ-539
END-EXEC
IF SQLCODE NOT EQUAL ZEROS
DISPLAY 'ERRO AO EXECUTAR SQL: ' SQLCODE
PERFORM 999-ABEND
END-IF
END-IF.
...
--
```

**1.3 Linha 07060000 a 07100000**

```cobol

CLOSE CICA600.

MOVE ZEROS TO SQLCODE.

PERFORM ECX-SQLCODE-PARA-FSTATUS.

MOVE ECX-WS-FSTATUS TO FS-TABA539.

```

- ****Erro Identificado:****

- O arquivo `TABA539` não é fechado explicitamente, mas o `FS-TABA539` é manipulado. Isso pode causar problemas em execuções subsequentes, pois o arquivo pode permanecer aberto.

- ****Solução Proposta:****

- Adicionar o fechamento explícito do arquivo `TABA539` para garantir que todos os arquivos sejam encerrados corretamente.

```cobol

CLOSE CICA600

TABA539.

MOVE ZEROS TO SQLCODE.

```

- --

**2. Coerência com COBOL/VSAM/DB2 Oracle**

- ****SQLCODE e File-Status:****

- A tentativa de mapear `SQLCODE` para `FS-TABA539` é incorreta.

`SQLCODE` é exclusivo para operações SQL e não deve ser usado para

manipular arquivos VSAM.

- A solução proposta remove essa incoerência, separando o tratamento de SQL e VSAM.

- ****Manipulação de Arquivos:****

- A abertura e fechamento de arquivos foram corrigidos para garantir que todos os arquivos sejam tratados adequadamente.

- ****Execução de SQL:****

- A instrução `EXEC SQL` está correta em termos de sintaxe, mas a lógica foi ajustada para garantir que variáveis sejam inicializadas antes do uso.

- --

**Resumo das Soluções Propostas**

1. ****Abertura de Arquivos:****

- Adicionar a abertura explícita do arquivo `TABA539` antes de manipular seu `FILE STATUS`.

2. ****Remoção de Redundâncias:****

- Eliminar a manipulação de `FS-TABA539` após operações SQL.

3. ****Inicialização de Variáveis:****

- Garantir que variáveis usadas em cláusulas `WHERE` estejam inicializadas.

4. ****Fechamento de Arquivos:****

- Adicionar o fechamento explícito do arquivo `TABA539` para evitar problemas em execuções subsequentes.

Se precisar de mais detalhes ou ajustes, é só avisar!