



Sistema de automação do desenvolvimento de sistemas

Eduardo Duarte Leôncio Silva, Faculdade Horizonte, Brasil
Michelli Ventura Borges, Faculdade Horizonte, Brasil
Ana Cristina Brandão Ribeiro Silva, Faculdade Horizonte, Brasil

RESUMO

Este trabalho buscou analisar a ferramenta SADS (Sistema de Apoio ao Desenvolvimento de Sistemas) demonstrando as vantagens da sua utilização no desenvolvimento de sistemas. A análise do caso foi realizada na empresa Spread Tecnologia, detentora dos direitos da respectiva ferramenta, e que concedeu a autorização para produzir este trabalho no seu ambiente organizacional com fins acadêmicos. Nesta perspectiva, com o desafio de apresentar uma solução que otimize o acesso, manipulação e integração de dados com suporte da linguagem Cobol, a empresa MSA-INFOR Sistemas e Automação Ltda produziu essa ferramenta visando contribuir significativamente para a eficiência e sucesso nos projetos de desenvolvimento de sistemas, que é utilizada para auxiliar no desenvolvimento e manutenção de sistemas, especialmente na plataforma Unisys. Assim, o estudo de caso apresentado proporciona uma compreensão aprofundada da respectiva tecnologia no desenvolvimento de sistemas, além de estimular a inovação e as habilidades essenciais para os profissionais de tecnologia da informação que contribuem para solucionar problemas complexos. Os resultados encontrados evidenciaram a eficácia do SADS na gestão eficiente de dados, modelagem avançada de sistemas e superação de desafios no processo de desenvolvimento. Desta forma, compreende-se que a proposta apresentada pelos profissionais autores deste estudo são adequadas para otimizar processos que envolvem sistemas, e promover uma gestão eficaz de dados, reforçando a relevância do SADS no cenário acadêmico e profissional.

Palavras-chave: Ferramenta SADS; Linguagem Cobol; Sistema.

1. INTRODUÇÃO

A pesquisa realizada sobre a ferramenta "SADS" (Sistema de Apoio ao Desenvolvimento de Sistemas) foi realizada na Spreads Tecnologia, empresa de grande porte, com mais de 2.000 funcionários, atua na área de tecnologia com sede em São Paulo, e filiais em Brasília e Rio de Janeiro, a qual desenvolve serviços de tecnologia e negócios digitais para mais de 300 grandes clientes de vários setores. Todavia, para este estudo o exemplo é de uma empresa/cliente do setor financeiro.

Silva, E.D.L., Borges, M.V., Silva, A.C.B.R.; Sistema de automação do desenvolvimento de sistemas. Revista de Empreendedorismo e Gestão de Micro e Pequenas Empresas V.9, N°1, p.101-124, Jan/Abr. 2024. Artigo recebido em 20/03/2024. Última versão recebida em 18/04/2024. Aprovado em 25/04/2024.

Nesse contexto, a pesquisa abordará como o uso desta ferramenta ajuda a melhorar a produtividade, automatizando e detectando erros na criação e manutenção de códigos fontes, aumentando assim a qualidade dos produtos desenvolvidos e padronizados conforme layout do cliente.

Além dos objetivos indicados, inclui-se adicionalmente explorar os benefícios e funcionalidades da respectiva ferramenta no desenvolvimento, versionamento e padronização dos sistemas desenvolvidos na linguagem COBOL.

Sendo assim, este trabalho busca analisar sobre como a ferramenta SADS beneficia as organizações que a utilizam, com o apoio dos objetivos específicos a seguir:

1. Avaliar a Eficácia - Compreender como o SADS impacta a eficácia das operações e tomada de decisões nas organizações;
2. Identificar Vantagens e Desafios - Analisar as vantagens que o SADS oferece, bem como os desafios enfrentados durante a implementação e uso;
3. Demonstrar Casos de Uso Bem-Sucedidos - Destacar exemplos específicos de uma organização que obteve sucesso ao implementar o SADS;
4. Analisar Funcionalidades Específicas - Explorar funcionalidades-chave do SADS e como elas atendem às necessidades específicas dos usuários.

Por fim, também serão abordadas questões de como a ferramenta facilita testes online nos ambientes de desenvolvimento, homologação e produção, sendo ela utilizada também como via de testes pelo cliente e equipes de homologação dos sistemas, ressaltando a tamanha importância que a utilização da ferramenta pode proporcionar.

A metodologia utilizada neste estudo de caso teve como base a coleta de dados de uma empresa de desenvolvimento de sistemas, onde a ferramenta SADS foi implementada, sendo realizadas análises ao longo do processo de desenvolvimento, comparando o uso da ferramenta com os métodos tradicionais.

No estudo de caso procurou-se destacar a aplicação da ferramenta SADS que é aumentar a eficiência e a qualidade na construção de sistemas, bem como reduzir os custos e o tempo de desenvolvimento. Para isso, a ferramenta destaca-se sendo abrangente e eficaz para profissionais envolvidos no desenvolvimento e manutenção de sistemas. As vantagens distintivas do SADS incluem:

1. Gerenciamento de Dados: Capacidade robusta para gerenciar dados internos e externos, facilitando o acesso e a manipulação eficiente de informações;
2. Modelagem Avançada: Funcionalidades avançadas de modelagem, permitindo uma representação precisa e visual dos sistemas em desenvolvimento;
3. Integração de Dados: Possibilidade de integração com fontes de dados diversas, proporcionando uma visão abrangente e integrada durante o processo de desenvolvimento;
4. Suporte a COBOL: Destaca-se no suporte à programação na linguagem COBOL, proporcionando uma solução abrangente para ambientes que utilizam essa linguagem.

2. ENQUADRAMENTO TEÓRICO

O desenvolvimento de sistemas envolve diversas etapas que exige conhecimento sobre as tecnologias e os aspectos técnicos do respectivo processo. Diante disso, cabe destacar a importância da compreensão sobre mainframe e a respectiva ferramenta visando maior entendimento sobre o estudo. Nesse caso, ressalta-se a exemplificação de Spread (2018, s.n) em que apresenta um case de um banco citando o uso da ferramenta, “fez a migração de toda sua plataforma para mainframe para ambiente IBM com a solução SADS”.

Além disso, é importante compreender o conceito da ferramenta na perspectiva de Spreads Tecnologia (2014, s.n)

O SADS é um conjunto completo e integrado de ferramentas que proporcionam alta produtividade no desenvolvimento e manutenção de sistemas nos ambientes mainframes IBM e Unisys, apoiando também o processo de implantação dos sistemas gerados e o planejamento e controle de suas execuções em produção, além de permitir e agilizar a modernização dos sistemas e automatizar a migração entre diferentes plataformas de hardware e software.

Por outro lado, considerando que a plataforma mainframe é parte da discussão do trabalho, cita-se a sua definição na visão de Bertolini *et al.*, (s.d) em que indica o seguinte: “o mainframe é um supercomputador, ou computador de grande porte, dedicado normalmente ao processamento de um volume grande de informações.” Os autores explicam sobre a usabilidade e funcionalidade desta plataforma, indicando seu maior uso nas áreas

Sistema de automação do desenvolvimento de sistemas

comerciais e científicas, em que exige uma grande estrutura para sua ocupação e funcionamento.

Tigre e Noronha (2013, p.117) por sua vez comentam que na década de 70, o mainframe foi desafiado com a criação dos microprocessadores. Porém, conforme os autores “Os mainframes eram alugados em pacotes de hardware, software e serviços e os fornecedores não tinham interesse em promover a obsolescência tecnológica de uma base instalada já amortizada, mas que ainda gerava grandes receitas.”

O Mainframe tem capacidade de operação em grandes volumes de dados com muita celeridade. Possuem muitas funcionalidades que podem promover resultados tanto para empresas de pequeno, médio e grande porte. Esta plataforma tem um histórico de evolução em sua tecnologia.

Além dos conceitos supracitados, considerando que a linguagem em questão se trata do Cobol, é importante descrever algumas afirmações sobre o assunto. A Microfocus (2020, p. 1) no artigo de Mike Madden, comenta que “Em uma era de pressão sem precedentes para inovar, em que as novas tendências parecem durar apenas pelos momentos mais breves, o sexagenário COBOL continua a brilhar”. Já no artigo de Sven Oldenburg na época quando foi projetado “tinha poucas linguagens de programação com que se preocupar” (Microfocus, 2020, p.5). e por outro lado, “(...) o conceito de portabilidade era simplesmente o requisito para dar suporte ao que se tornou conhecido como computadores mainframe, que eram a única opção de computação no momento.”

Enfim, o compreender a linguagem Cobol é importante para o entendimento da proposição neste estudo de caso. Nesta perspectiva cabe destacar as afirmações e contextualizações de Bauer et al., (s.d, p.9), em que compreende tratar-se de: “Uma linguagem de programação de computador que foi projetada especificamente para negócios, Common Business-Oriented Linguagem, COBOL.” Assim, conforme observado nas afirmações essa linguagem é a responsável constante das execuções operacionais da rotina confiável do universo financeiro, tendo em vista que os dados mais críticos de algumas empresas são processados por ela. Além disso, “(...) tem transformado e apoiado negócios globalmente desde a sua invenção em 1959.”

3 METODOS DE PROCEDIMENTOS DO ESTUDO DE CASO

O estudo na busca de informações para elaboração deste trabalho teve a pesquisa de fatos sendo conduzida por meio do desenvolvimento e manutenção dos sistemas DCB e OCG realizados pela ferramenta SADS, visando demonstrar os benefícios do uso da ferramenta e buscando levantar conhecimentos relacionados ao tema a ser explorado, além das experiências no ambiente real.

Além disso, cabe destacar a visão de Gil (2010) em que o estudo de caso tem a peculiaridade de aprofundamento em um determinado foco ou poucos objetos de tal forma a promover informações e conhecimentos ampliados e explicativos do respectivo.

Os métodos de abordagens utilizados nesta pesquisa, foram os manuais disponibilizados pela empresa e o conhecimento tácito obtido por meio dos anos trabalhados com essa ferramenta. Nesta perspectiva a coleta de dados se deu com a respectiva ferramenta e a documentação dos manuais disponibilizados pela empresa.

Os participantes da pesquisa foram os analistas e desenvolvedores do sistema DCB e do sistema OCG.

Inicialmente se fez uma leitura dos manuais discutindo-se sobre os aspectos técnicos e suas principais funcionalidades, em seguida aplicou-se a ferramenta em um exemplo prático de geração de código.

3.1 Requisitos de hardware e software para utilizar a ferramenta SADS

Para utilização do SADS Versão 7.3, as seguintes configurações de hardware e software devem ser:

- CPU: Compatível com z/OS 2.1;
- Software: Sistema Operacional z/OS 2.1 ou superior;
- Gerenciador de transações: Cics 4.2 ou superior;
- Banco de Dados: DB2 versão 10 ou superior;
- Enterprise Cobol versão 4.2 ou superior.

O SADS gera programas seguindo lógicas padronizadas que atendem à maioria das necessidades de aplicativos online e batch. Quanto mais simples a lógica e quanto mais próxima de uma lógica padronizada estiver a função a ser executada pelo programa, maior será a capacidade de geração do SADS.

- Programas simples e funcionalmente dentro dos padrões são gerados completamente a partir de um conjunto mínimo de especificações;

- Programas mais complexos e/ou que possuam peculiaridades funcionais têm seu cadastramento complementado pelo uso de módulos de processamento.

3.2 Tipos de navegações entre as telas

O SADS tem à disposição duas formas de navegação para os sistemas aplicativos:

- Navegação via host;
- Navegação interpretativa.

3.3 Navegação via host

, em que consta os itens do projeto conceitual, projeto lógico, projeto físico, A navegação via host é aquela em que o usuário solicita as transações diretamente a partir dos menus de opção, informando neles o número da opção e os argumentos referentes à transação que ele deseja executar.

Na Figura 1, acessa-se o menu de consulta da estrutura de dados através da navegação via host, podemos fazer isso digitando o comando 'D CONESTDAD' na barra de ação, onde a letra 'D' significa uma solicitação de desvio para a tela a ser informada, neste caso a tela de estrutura de dados.

Assim, observa-se os itens do projeto conceitual, projeto lógico, projeto físico, em que estão indicados em cada coluna da figura a seguir.

Figura 1

Ação a ser realizada na linha de comando

Sistema de automação do desenvolvimento de sistemas

```
SDD0.000          SADS - SISTEMA DE AUTOMACAO DO DESENVOLVIMENTO DE SISTEMAS
ACAO      D_CONESTDAD
----- Inicio      Anterior      Origem      Desvio      Fim -----
PROJETO CONCEITUAL          PROJETO LOGICO          PROJETO FISICO
Sistema                    TELA                    DATaset
ENTidade                  TABela SADS             ARQuivo
DOMinio                   CONSistencia            USO geral
ATRibuto e sinonimo      RELatorio              AREa de trabalho
                          HOST e menu            SUB-rotina
                          CAMpo e sinonimo      PRograma
                                                   LIBrary de procedure
                                                   TABelas RELacionais
                          FUNCOES DE APOIO
                          UTILitarios
                          INFormativos
ESCOLHA _____
F1=HELP F3=SAIR F4=ORIG F5=ANT CL=FIM
4 B TD11733 002/022
```

Nota: autoria própria.

A seguir observa-se que na Figura 2 encontra-se o resultado do comandindo direto para a tela solicitada em que a ação foi comandada.

Figura 2

Resultado da solicitação de desvio na navegação via host.

```
SDD0.127          CONESTDAD - ESTRUTURA DE DADOS          24/11/23 15:04:43
ACAO
----- Inicio      Anterior      Origem      Desvio      Fim -----
Estrutura ... _____
F1=HELP F3=SAIR F4=ORIG F5=ANT CL=FIM
4 B TD11733 009/025
```

Nota: autoria própria.

3.3.1 Navegação interpretativa

A navegação interpretativa é o processo que permite caminhar pelas funções disponíveis de um ou vários sistemas, através de um deslocamento amigável entre seus menus de escolha, menus de operação e telas de processamento.

Sistema de automação do desenvolvimento de sistemas

Precisamos utilizar os menus de escolha informando qual menu desejamos acessar. As opções aparecem na tela e devemos informar apenas as letras que aparecem em caixa alta.

Para acessar a mesma tela que utilizamos como exemplo na navegação anterior devemos informar na linha de escolha a função de apoio 'INF' (Informativos) como consta na Figura abaixo.

Figura 3

Informativos através do menu de escolha.

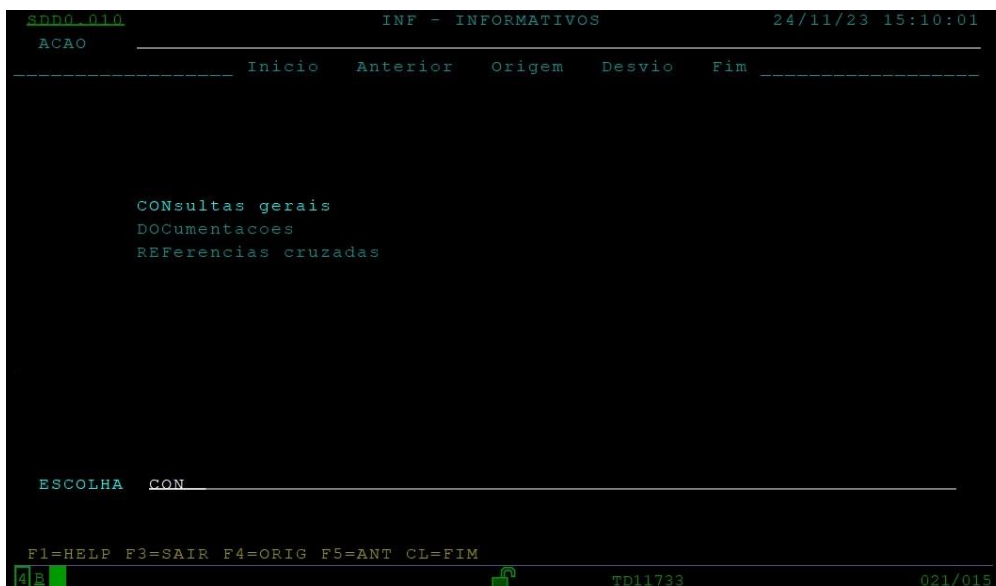
```
SDD0.000          SADS - SISTEMA DE AUTOMACAO DO DESENVOLVIMENTO DE SISTEMAS
ACAO
----- Inicio  Anterior  Origem  Desvio  Fim -----
PROJETO CONCEITUAL      PROJETO LOGICO      PROJETO FISICO
Sistema                 TELa                 DATaset
ENTidade                TABela SADS         ARQuivo
DOMinio                 CONSistencia        USO geral
ATRibuto e sinonimo    RELatorio           AREa de trabalho
                        HOST e menu         SUB-rotina
                        CAMpo e sinonimo    PROgrama
                        LIBrary de procedure
                        TABelas RELacionais
                        FUNCOES DE APOIO
                        UTIlitarios
                        INFormativos
ESCOLHA  INF
F1=HELP F3=SAIR F4=ORIG F5=ANT CL=FIM
4  E  TD11733  021/015
```

Nota: autoria própria.

Após isso devemos informar no menu de escolha que queremos acessar as consultas gerais, conforme Figura 4.

Figura 4

Consultas Gerais através do menu de escolha.

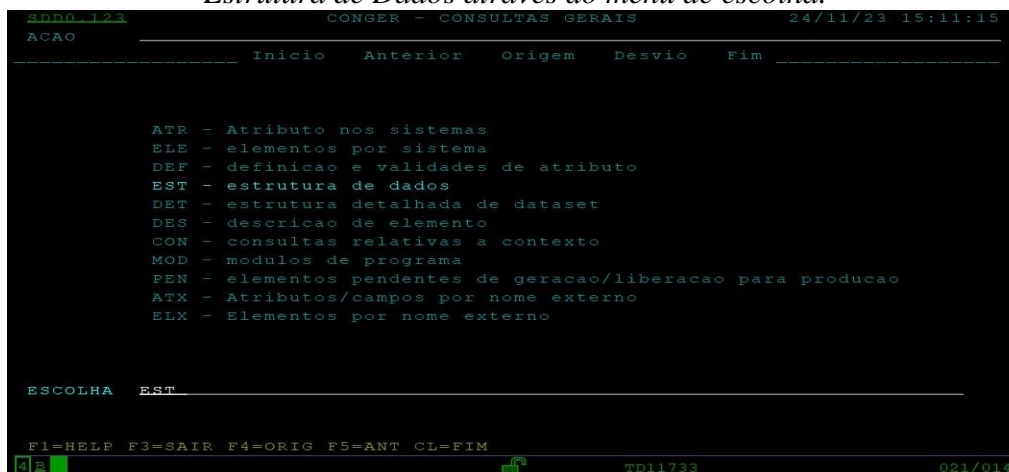


Nota: autoria própria.

Por último acessa-se a tela de estrutura de dados, informando o menu 'EST' como mostrado na Figura 5.

Figura 5

Estrutura de Dados através do menu de escolha.



Nota: autoria própria.

Assim encontra-se a estrutura de dados esperada, conforme apresentado na figura a seguir.

Figura 6

Estrutura de Dados acessada via menu de escolha.

```
SDD0.127          CONESTDAD - ESTRUTURA DE DADOS          24/11/23 15:12:16
ACAO
----- Inicio   Anterior  Origem   Desvio   Fim -----
 
Estrutura ... _____

F1=HELP F3=SAIR F4=ORIG F5=ANT CL=FIM
4 B TD11733 009/025
```

Nota: autoria própria.

Este método de navegação, por ser orientado por menus, amplia a facilidade de utilização do sistema, tornando-o mais amigável e evitando a memorização dos *trancodes* nele utilizados.

3.3.2 Padronização

O uso de padrões auxilia na automação de procedimentos e traz como principais vantagens a simplificação nos procedimentos de desenvolvimento/manutenção de sistemas e a normalização operacional dos produtos desenvolvidos.

O SADS possui padrão para três grandes itens relacionados ao ambiente computacional:

- Elementos do sistema;
- Lógica de programas;
- Interface com o usuário.

Os programas fonte gerados pelo SADS são escritos em Cobol e seguem rigorosamente as técnicas de programação estruturada, sendo de fácil leitura e compreensão. O processo de desenvolvimento dos programas pode ser resumido nas seguintes tarefas:

- Cadastramento do programa no repositório;
- Geração do programa;

Sistema de automação do desenvolvimento de sistemas

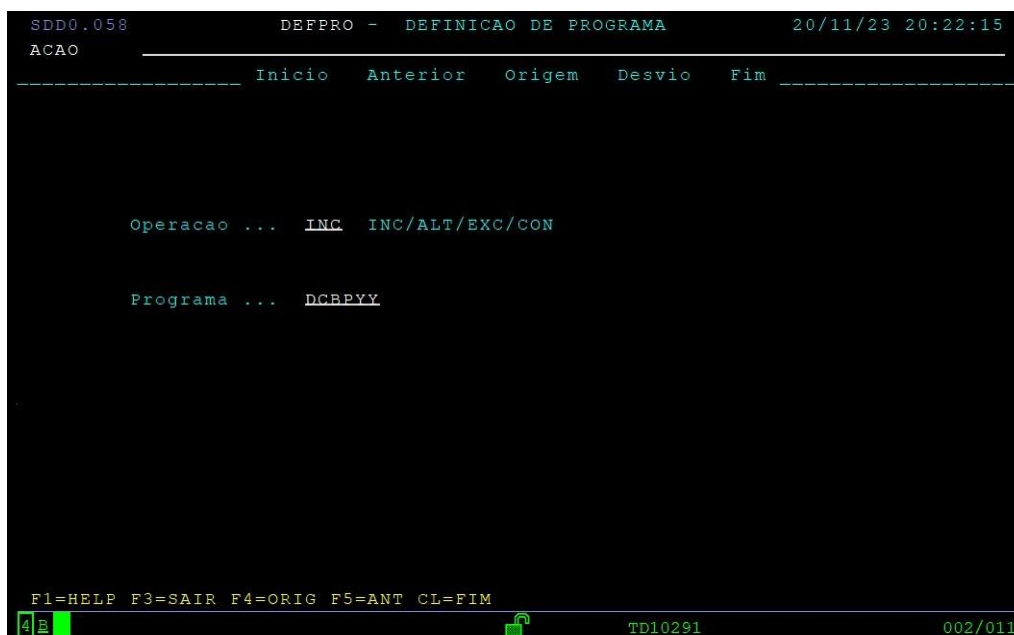
- Compilação;
- Testes.

3.3.3 Cadastramento do programa no repositório

Para incluir um programa no repositório deve-se ir a tela de definição de programa e informar a operação de inclusão, passando o nome do programa a ser utilizado, conforme demonstrado a seguir.

Figura 7

Inclusão de novo programa no repositório.



Nota: autoria própria.

Após definido o programa a ser incluído iremos informar o tipo de lógica e outras informações que já possuem seus campos pré-preenchidos por padrão (Figura 8).

Figura 8

Definição do tipo de lógica do programa.

```
SDD2_061  E2  MSA      Definicao de Programa/Sub-rotina      20/11/23 20:23:09

Programa DCBPYY                Geracao completa pelo SADS (S/N).....:  S
Nativo IBM(I)/Convertido(C) ou Unisys(U) .... I
Gerar para CICS (S/N) .....:  N  Cursores com clausula WITH HOLD (S/N) :  S
Executar mesmo com registro em processo de atualizacao (S/N) :  S
Consultas: Simples ..... CO    Atualizacoes: Simples ..... AT
          Paginada ..... CP      Paginada ..... AP
          Com encadeamento ..... CE
Emissoes:  Em Impressora Remota .. ER  Sub-rotinas: Padrao ..... SR
                                                Consistencia Geral.. SC
                                                Exc. Itens Tabela .. ST

Logica de Programas Batch ..... BT    Genericas:  De Selecao ..... SE
                                                Livre ..... LL

Tipo de Logica BT

Nome ..... PROGRAMA_EXEMPLO
Analista ..... ANALISTA_XXXXXX
Programador ..... PROGRAMADOR_YYYYY
Usuario ..... USUARIO_12345

Operacao INC

F1=HELP F3=SAIR F4=MENU CL=LOGOFF

4E [lock icon] TD10291 003/078
```

Nota: autoria própria.

Na próxima tela deverá ser informado os recursos que serão utilizados no programa como tabelas DB2, arquivos, sub-rotinas e outros.

Na Figura 9 definimos como exemplo que o programa deverá possuir uma tabela ‘DCBB01’, um arquivo de saída ‘DCBAD2’ e acessará a sub-rotina ‘SSRS02’.

Figura 9

Definição de recursos do programa.

```

SDD2.062  E3  MSA      DEFINICAO DE RECURSOS      25/11/23 15:37:10
DCRPPY  PROGRAMA
Elemento Cpl Elemento Cpl Elemento Cpl Elemento Cpl Elemento Cpl
Tabela DB2      DCBB01
Arquivo I/O/U   DCBAD2
Lib. W/P/C/I/M
Tela .....
Sub-rotina      SRS02
Tab. SADS
Relatorio.
Reg. Geral

Use DIVERSOS para cadastrar qualquer dos recursos que excederem a 5 elementos
DIVERSOS 1
..... 2
..... 3
..... 4
..... 5
..... 6
..... 7
Usuario DB2 das tabelas SBGB02 / SBGB01
OPERACAO ALT

F1=HELP F3=SAIR F4=MENU CL=LOGOFF
4 B TD11950 004/017
    
```

Nota: autoria própria.

Em seguida é apresentada a tela para que sejam definidas as variáveis de *work*, ou seja, as variáveis utilizadas no processamento do programa (Figura 10).

Figura 10

Definição das variáveis da área de trabalho.

```

SDD2.063  E4  MSA      ESTRUTURA DE TRABALHO (AREA DE WORK)  20/11/23 20:49:36
PROGRAMA/SUB-ROTINA  DCRPPY
SEQ  ATRIB/CAMPO  SUFIXO  NIVEL  NDEF  SIS  COMP  FORM  TAM  DEC  SINAL  VAL  OCCURS
01  0010  DATAPARM  ___  01  P  ___  ___  ___  ___  ___  ___  ___
02  0020  AREA-WORK  ___  01  P  ___  ___  ___  ___  ___  ___  ___
03  0030  DATA  ___  10  P  ___  ___  ___  ___  ___  ___  ___
04  0040  VALOR  ___  10  P  ___  ___  ___  ___  02  ___  ___  ___
05  0050  TEXTO  ___  10  P  ___  ___  ___  ___  ___  ___  ___
06  0060  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___
07  0070  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___
08  0080  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___
09  0090  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___
10  0100  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___
11  0110  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___
12  0120  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___
13  0130  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___
14  0140  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___
15  0150  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___  ___

CONTINUA(S/N) N OPERACAO ALT

F1=HELP F3=SAIR F4=MENU CL=LOGOFF
4 B TD10291 006/011
    
```

Nota: autoria própria.

Na Figura 11 são definidas as estruturas de dados para que o próprio SADS gere as movimentações automaticamente entre as estruturas, neste exemplo será realizada a movimentação da tabela DCBB01, definida como sendo a entrada de dados, para o arquivo de saída DCBAD2.

Figura 11

Definição das estruturas de dados.

```
SDD2_034  J4 MSA  DEFINICAO DE PROCESSAMENTO - BATCH  20/11/23 20:41:46
          LOGICA BATCH DE PROGRAMA NATIVO IBM

PROGRAMA ....: DCRPYY

PARAMETRO ...: _____

                                     Atualizacao DB2
          Estrutura      Tipo      Ultimo      I=Incluir
          DB2/Arquivo    Set Acesso  Atributo    A=Alterar
          Relatorio      Arg. F/W    Pesquisa    E=Excluir
          -----      -
ENTRADA .....: DCBR01_____

SAIDA .....: DCBAD2_____

                                OPERACAO ALT

F1=HELP F3=SAIR F4=MENU CL=LOGOFF
4B TD10291 006/040
```

Nota: autoria própria.

Com todas as definições do programa prontas o SADS apresenta a estrutura do programa com os módulos/parágrafos que serão gerados (Figura 12), finalizando a parte de definição do programa.

Figura 12

Estrutura dos módulos a serem gerados.

```

STEP5_001  F2  MSA  SELECAO DE MODULO PARA PROGRAMA  20/11/23 20:43:44
SDD02_72                                     PAGINA .: 01
----- MODULOS ----- .PROGRAMA/SUB-ROTINA DCRPYY LOGICA BT
INICIO      FIM      EXC
01          -        -        -        .Inicio
02  111  N    116  N    -        ....Iniciais
03  151  N    156  N    -        ....Abrir-banco
04  171  N    176  N    -        ....Abrir-arquivo1
05          -        -        -        ....Ler-regl
06  181  N    186  N    -        ....Tratar-header1
07  211  N    216  N    -        ....Formatar1
08  221  N    226  N    -        .....Compor-reg-saidal
09  711  N    716  N    N        .....Gravar-reg-saidal
10          -        -        -        .....Ler-regl
11  751  N    756  N    -        ....Finais
12  761  N    766  N    -        ....Fechar-banco
13  771  N    776  N    -        ....Fechar-arquivo2
14          -        -        -        .Fim
15          -        -        -
16  611  N    616  N    N        .Ler-regl

OPERACAO I CONTINUA(S/N) N PROXIMA PAG. 99
PF1=HELP PF3=VOLTAR CLEAR=SAIR PF7=RETORNAR PF8=AVANCAR
4E TD10291 003/050
    
```

Nota: autoria própria.

3.3.4 Geração do programa

Após realizadas as devidas alterações do código fonte através da ferramenta SADS é necessário gerar o programa, para isso tem a opção “Geração” no menu de escolha localizada em funções genéricas (Figura 13).

Figura 13.

Opção de geração de programa.

```

SDD0_057  PRO - ESPECIFICACAO DE PROGRAMA  20/11/23 20:45:33
ACAO
----- Inicio Anterior Origem Desvio Restaura Fim
NATIVO IBM/CONVERTIDO/LEGADO/MANUAL
DEFINICAO (Identificacao) RECURSOS (Relacionamentos)
AREA de work (Variaveis de trabalho) LOGICA de programa
QUEbra/classificacao SELEcao de modulos

NATIVO IBM/CONVERTIDO/MANUAL LEGADO
CADASTRAMENTO de modulo IBM CADASTRAMENTO de modulo
CADASTRAMENTO de SQL/arquivo RESEQUENCIAMENTO de modulo
EDICAO de SECOES da IDENTIFICATION EDICAO cobol
CONverte PROGRAMA para IBM

FUNCOES GENERICAS
GERacao DESCRICAO
COMpilacao/binder DUPLICACAO de programa/modulo
EXPORTACAO/IMPORTACAO - mod/sql/edit
IMPORTACAO de programa COBOL manual

ESCOLHA GER
Projeto Fisico

F1=HELP F3=SAIR F4=ORIG F5=ANT CL=FIM
4E TD10291 021/015
    
```

Nota: autoria própria.

Sendo assim, será necessário apenas informar o nome do programa e a ferramenta se encarrega de gerar o código conforme apresentado a seguir (Figura 14).

Figura 14

Informando o programa a ser gerado.

```
SDRN_064          GERPRO - ON-LINE E BATCH          20/11/23 20:46:09
ACAO
----- Inicio  Anterior  Origem  Desvio  Fim
-----
Programa .....: DCBPVY
-----
Projeto Fisico
-----
F1=HELP F3=SAIR F4=ORIG F5=ANT CL=FIM
TD10291 002/011
```

Nota: autoria própria.

A compilação do programa é disparada pela ferramenta, mas a visualização do resultado da compilação e a realização dos testes são possíveis apenas através do TSO/ISPF.

O SADS possibilita aos usuários visualizarem a lógica do programa a ser gerada, exibindo todos os seus parágrafos em seus respectivos níveis hierárquicos, demonstrado anteriormente na Figura 12.

Através desta visualização estrutural dos parágrafos, o usuário pode selecionar módulos para complementação ou alteração do código fonte do programa a ser gerado. Na Figura 15 estamos informando que queremos fazer uma intervenção no final do parágrafo ‘Compdor-reg-saida1’.

Por padrão os campos de início e fim de módulo estão preenchidos com ‘N’, podendo ser alterado para ‘L’ caso se deseje inserir instruções no início ou fim do módulo, ou alterado para ‘T’ se o objetivo for trocar o módulo por completo.

Figura 15

Incluindo comando no final do parágrafo.


```

STP5_001  F2  MSA      SELECAO DE MODULO PARA PROGRAMA      20/11/23 20:57:33
SDD02_72                                     PAGINA  : 01
----- MODULOS ----- .PROGRAMA/SUB-ROTINA DCRPYY      LOGICA BT
      INICIO      FIM      EXC
01      -      -      -      .Inicio
02  111  N      116  N      -      ...Inicioais
03  151  N      156  N      -      ...Abrir-banco
04  171  N      176  N      -      ...Abrir-arquivo1
05      -      -      -      ...Ler-reg1
06  181  N      186  N      -      ...Tratar-header1
07  211  N      216  N      -      ...Formatar1
08  221  N      226  L      -      .....Compor-reg-saidal
09  711  N      716  N      N      .....Gravar-reg-saidal
10      -      -      -      .....Ler-reg1
11  751  N      756  N      -      ...Finais
12  761  N      766  N      -      ...Fechar-banco
13  771  N      776  N      -      ...Fechar-arquivo2
14      -      -      -      .Fim
15      -      -      -      -
16  611  N      616  N      N      .Ler-reg1

      OPERACAO I      CONTINUA(S/N) N      PROXIMA PAG. 99

PF1=HELP PF3=VOLTAR CLEAR=SAIR PF7=RETORNAR PF8=AVANCAR
  4  E  TD10291      002/002
    
```

Nota: autoria própria.

Módulos de processamento são trechos de programas que podem ser cadastrados pelo usuário moldando o código fonte gerado às suas necessidades, incluindo, alterando e substituindo trechos do código fonte que seriam normalmente gerados pelo SADS.

Esses trechos serão compostos por instruções Cobol e comandos de módulo que funcionam como verdadeiras macros instruções, fatorando seu trabalho a seu tempo. Com uma única linha de especificação pode-se:

- Realizar acesso a *datasets*, incluindo tratamento de erro padronizado;
- Montar loops estruturados;
- Fazer acesso a tabelas, sub-rotinas;
- Emitir mensagens, de forma padronizada;
- Movimentar dados entre estruturas etc.

3.3.5 Resultados

A seguir serão demonstrados alguns casos de geração de código fonte através de intervenções no código e/ou geração padrão da ferramenta SADS.

3.3.5.1 Chamada a sub-rotina

Uma sub-rotina normalmente é criada com um trecho de código fonte a ser utilizado por vários programas, para que não seja necessário colocar o mesmo trecho de código em cada um dos programas se cria uma sub-rotina com este código e no programa apenas se faz a chamada desta sub-rotina informando os parâmetros necessários para a sua execução; seria

Sistema de automação do desenvolvimento de sistemas

como se um programa estivesse chamando outro programa para fazer determinada ação específica.

Na Figura 16 temos a forma de como definir na ferramenta SADS a chamada a sub-rotina ‘SSRS02’, apenas colocando a palavra reservada ‘CALL’ e o nome da sub-rotina.

Figura 16

Definindo chamada a sub-rotina.

```
SDD2_031  J1 MSA      DEFINICAO DE MODULO DE PROCESSAMENTO  20/11/23 20:59:21
ESTE MODULO SERA INSERIDO NO FINAL DA ROTINA COMPOR-BEG-SAIDA1
ELEMENTO DCBPYY      MODULO LINEAR_  226      SEQUENCIA 000100
(TAM. MAX. CAMPOS#) - COMANDOS EM COBOL#66(7-72) OU COMANDOS DE MODULOS DO SADS
COMANDO#12 TP/ID#8 ORIGEM#10 DESTINO#10 SET#2 ULT. ATR#11 INDICE#1 V/FORMULA #2
1-----*A-1-B--+-2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----72
01 _____
02 _____
03 CALL SSRS02
04 _____
05 _____
06 _____
07 _____
08 _____
09 _____
10 _____
11 _____
12 _____
13 _____
14 _____
15 _____
OPERACAO I          CONTINUA (S/N) N          PROXIMA SEQUENCIA 000000
F1=HELP F3=SAIR F4=MENU CL=LOGOFF F5=EXP F6=IMP
TD10291 007/011
```

Nota: autoria própria.

Após realizada esta intervenção no código e gerado novo programa com esta atualização, podemos verificar o resultado no código fonte (Figura 17). Neste código gerado a ferramenta criou a chamada a sub-rotina, mas deixou informado que o usuário deverá realizar o teste de erro no retorno do acesso a sub-rotina.

Figura 17

Código gerado após inclusão de chamada a sub-rotina.

Sistema de automação do desenvolvimento de sistemas

```
102200                                     S.A.D.S.
162488*****-----S.A.D.S.
162588 22088-00-COMPOR-REG-SAIDA1.         S.A.D.S.
162688                                     S.A.D.S.
162788      MOVE B0100-HDRDATA             TO BD201F-HDRDATA.     S.A.D.S.
162888      MOVE B0100-HDRHORA             TO BD201F-HDRHORA.     S.A.D.S.
162988      MOVE B0100-HDRESTACAO          TO BD201F-HDRESTACAO. S.A.D.S.
163088      MOVE B0100-HDRPROGRAMA         TO BD201F-HDRPROGRAMA. S.A.D.S.
163188      MOVE B0100-HDRSTATGER          TO BD201F-HDRSTATGER. S.A.D.S.
163288      MOVE B0100-ASBCODBANCO        TO BD201F-ASBCODBANCO. S.A.D.S.
163388                                     S.A.D.S.
163488                                     S.A.D.S.
163588                                     S.A.D.S.
163688$SET OMIT = CICS                     &LIN-226
163788* SUBROTINA DE DATAS                 &LIN-226
163888      CALL "SSRS02" USING SSR02-DATAPARM. &LIN-226
163988*      TESTAR A CONDICAO DE ERRO DE ACESSO A SUBROTINA &LIN-226
164088$POP OMIT                            &LIN-226
164188                                     S.A.D.S.
164288                                     S.A.D.S.
164388                                     S.A.D.S.
164488 22088-99-EXIT.                      S.A.D.S.
164588      EXIT.                          S.A.D.S.
164688                                     S.A.D.S.
164788*****-----S.A.D.S.
-----
```

Nota: autoria própria.

Um detalhe a ser observado na figura anterior é que as intervenções realizadas no código são destacadas para que seja possível identificar o que foi gerado por padrão pela ferramenta e o que foi incluído pelo usuário. Neste caso, informamos na Figura 15 que iríamos fazer uma intervenção no final do parágrafo ‘Compor-reg-saida1’. O código padrão gerado pela ferramenta sempre aparecerá com a palavra ‘S.A.D.S.’ na coluna 80.

3.3.5.2 Acesso a base de dados (Tabela DB2)

O acesso a uma base de dados pode ter como finalidade apenas uma consulta, uma inclusão, alteração ou exclusão de registros da base. Na figura 18 está sendo demonstrado um acesso para realizar uma consulta na base de dados ‘DCBB01’. É informado o comando SQL a ser realizado (SQLSELECT), um alias (apelido) para este comando, o nome da base a ser acessada e o índice que possui as chaves já definidas (uma base pode conter vários índices diferentes para otimização do acesso).

Figura 18

Definindo o acesso a base de dados.

Sistema de automação do desenvolvimento de sistemas

```
SDD2_031 J1 MSA DEFINICAO DE MODULO DE PROCESSAMENTO 20/11/23 21:13:16
ESTE MODULO SERA INSERIDO NO FINAL DA ROTINA COMPOR-REG-SAIDA1
ELEMENTO DCBPYV MODULO LINEAR 226 SEQUENCIA 000100
(TAM. MAX. CAMPOS#) - COMANDOS EM COBOL#66(7-72) OU COMANDOS DE MODULOS DO SADS
COMANDO#12 TP/ID#8 ORIGEM#10 DESTINO#10 SET#2 ULT. ATR#11 INDICE#1 V/FORMULA #2
1-----*A-1-B-+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----72
01 _____
02 _____
03 CALL SSRS02
04 _____
05 _____
06 SQLSELECT PRG01 DCBB01 A
07 _____
08 _____
09 _____
10 _____
11 _____
12 _____
13 _____
14 _____
15 _____
OPERACAO A CONTINUA (S/N) N PROXIMA SEQUENCIA 000000
F1=HELP F3=SAIR F4=MENU CL=LOGOFF F5=EXP F6=IMP
4] TD10291 007/011
```

Nota: autoria própria.

Realizando a geração do código pode-se identificar que foi incluída a chamada a um novo parágrafo (Figura 19), onde estará o comando de acesso a base DCBB01.

Figura 19

Parágrafo criado após inclusão de acesso a tabela DB2.

```
-----S.A.D.S.
162488*****-----S.A.D.S.
162588 22088-00-COMPOR-REG-SAIDA1.S.A.D.S.
162688 S.A.D.S.
162788 MOVE B0100-HDRDATA TO BD201F-HDRDATA.S.A.D.S.
162888 MOVE B0100-HDRHORA TO BD201F-HDRHORA.S.A.D.S.
162988 MOVE B0100-HDRESTACAO TO BD201F-HDRESTACAO.S.A.D.S.
163088 MOVE B0100-HDRPROGRAMA TO BD201F-HDRPROGRAMA.S.A.D.S.
163188 MOVE B0100-HDRSTATGER TO BD201F-HDRSTATGER.S.A.D.S.
163288 MOVE B0100-ASBCODBANCO TO BD201F-ASBCODBANCO.S.A.D.S.
163388 S.A.D.S.
163488 S.A.D.S.
163588 S.A.D.S.
163688$SET OMIT = CICS &LIN-226
163788* SUBROTINA DE DATAS &LIN-226
163888 CALL "SSRS02" USING SSR02-DATAPARM. &LIN-226
163988* TESTAR A CONDICAO DE ERRO DE ACESSO A SUBROTINA &LIN-226
164088$POP OMIT &LIN-226
164188 S.A.D.S.
164288 S.A.D.S.
164388 PERFORM 995001-00-SELECT-DCBB01 &LIN-226
164488 THRU 995001-99-EXIT. &LIN-226
164588 S.A.D.S.
164688 S.A.D.S.
164788 S.A.D.S.
164888 22088-99-EXIT. S.A.D.S.
164988 EXIT. S.A.D.S.
165088 S.A.D.S.
165188*****-----S.A.D.S.
-----S.A.D.S.
```

Nota: autoria própria.

Neste novo parágrafo criado teremos o comando SELECT na base DCBB01 utilizando como chave o índice informado. Também é gerado o tratamento de SQLCODE para este acesso (Figura 20).

Figura 20

Geração do comando Select e tratamento de Sqlcode.

```
212888      , :BB0100-TITLNKQR S.A.D.S.
212988      , :DCBB0103.SEDIDLCK S.A.D.S.
213088      FROM DCBB00.DCBB01 S.A.D.S.
213188      WHERE S.A.D.S.
213288      B0100_TITCODFILL = S.A.D.S.
213388      :B01A-TITCODFILL S.A.D.S.
213488      AND B0100_CTCCHVCONTA = S.A.D.S.
213588      :B01A-CTCCHVCONTA S.A.D.S.
213688      AND B0100_CATCODTIT = S.A.D.S.
213788      :B01A-CATCODTIT S.A.D.S.
213888      AND B0100_TITNUM = S.A.D.S.
213988      :B01A-TITNUM S.A.D.S.
214088      AND B0100_ASBCODBANCO = S.A.D.S.
214188      :B01A-ASBCODBANCO S.A.D.S.
214288      AND B0100_TITDGVCHASB = S.A.D.S.
214388      :B01A-TITDGVCHASB S.A.D.S.
214488      END-EXEC. S.A.D.S.
214588      IF SQLCODE IN SQLCA EQUAL 0 S.A.D.S.
214688      PERFORM 97555-00-BCO-UWA-DCBB01 S.A.D.S.
214788      THRU 97555-99-BCO-UWA-DCBB01 S.A.D.S.
214888      END-IF. S.A.D.S.
214988      S.A.D.S.
215088      IF SQLCODE IN SQLCA EQUAL 100 S.A.D.S.
215188      MOVE "S" TO NOTFOUND-DCBB01 S.A.D.S.
215288      GO TO 995001-99-EXIT S.A.D.S.
215388      END-IF. S.A.D.S.
215488      IF SQLCODE IN SQLCA LESS 0 S.A.D.S.
215588      DISPLAY "EXCECAO EM - 995001-00-SELECT-DCBB01" S.A.D.S.
215688      GO TO 98000-00-EXCECAO-FATAL S.A.D.S.
215788      END-IF. S.A.D.S.
215888      S.A.D.S.
215988      995001-99-EXIT. S.A.D.S.
216088      EXIT. S.A.D.S.
216188      S.A.D.S.
216288      *****-----S.A.D.S.
```

Nota: autoria própria.

Estes foram alguns exemplos simples de intervenções no código padrão gerado pela ferramenta SADS.

4 CONSIDERAÇÕES FINAIS

Durante o estudo de caso, identificaram-se diversos desafios que foram superados com a implementação do SADS:

1. **Gestão Ineficiente de Dados:** O SADS proporcionou uma gestão eficiente de dados, superando desafios relacionados à manipulação e integração de informações, permitindo uma visão integrada durante o desenvolvimento;
2. **Modelagem Complexa de Sistemas:** Os recursos avançados de modelagem do SADS foram fundamentais para superar desafios na representação precisa e visual de sistemas complexos, facilitando o entendimento durante o processo de desenvolvimento;
3. **Integração de Dados em Tempo Real:** O SADS contribuiu para a superação de desafios relacionados à obtenção de dados atualizados, possibilitando a integração em tempo real, o que foi crucial para tomadas de decisões informadas;

4. Suporte à Linguagem COBOL: Em ambientes onde a linguagem COBOL é predominante, o SADS se destacou ao oferecer suporte robusto a essa linguagem, facilitando a adaptação e integração em projetos legados;
5. Eficiência no Desenvolvimento: A ferramenta contribuiu para aumentar a eficiência no desenvolvimento de sistemas, otimizando processos e fornecendo ferramentas avançadas para enfrentar desafios específicos do ciclo de vida do desenvolvimento de software.

As limitações da ferramenta SADS foram identificadas durante o estudo de caso:

1. Curva de Aprendizado: O SADS pode apresentar uma curva de aprendizado para usuários iniciantes, exigindo tempo para se familiarizarem completamente com suas funcionalidades;
2. Compatibilidade com Outras Linguagens: Apesar do suporte robusto à linguagem COBOL, a compatibilidade com outras linguagens pode ser limitada, o que pode ser uma consideração relevante em ambientes heterogêneos;
3. Customização Limitada: A capacidade de customização da ferramenta pode ser limitada em comparação com soluções mais flexíveis, o que pode restringir a adaptação a requisitos específicos;
4. Dependência de Infraestrutura Legada: Em ambientes onde a infraestrutura legada é significativa, a dependência do SADS pode criar desafios de integração com tecnologias mais recentes;
5. Atualizações e Suporte: A disponibilidade de atualizações e suporte contínuo pode ser uma limitação, impactando a capacidade de lidar com requisitos emergentes e tecnologias mais recentes.

Os resultados obtidos com a implementação da ferramenta SADS foram altamente positivos. Houve uma redução significativa no tempo de desenvolvimento de projetos, uma vez que a geração automática de código agilizou o processo e reduziu a necessidade de escrever códigos manualmente. Além disso, a detecção precoce de erros e a automatização de testes contribuíram para a melhoria da qualidade do software entregue.

Portanto, este estudo de caso demonstrou que a ferramenta SADS foi capaz de promover ganhos expressivos no desenvolvimento de sistemas, tanto em termos de eficiência

Sistema de automação do desenvolvimento de sistemas

como de qualidade. A adoção dessa ferramenta pode se tornar uma estratégia poderosa para empresas que buscam otimizar seus projetos de software e se manter competitivas em um mercado cada vez mais exigente.

REFERÊNCIAS

Bauers, *et al.* Open Mainframe Project: Um guia para iniciantes em COBOL usando ferramentas modernas, Programação Cobol com VSCode. (2023) Disponível em:> file:///E:/ARTIGOS/COBOL-Visual_Studio.pdf. (Manual técnico).

Bertolini, R. B, Sasaki, E. N. (s.d). A História dos Computadores e Mainframes nos Ambientes Comerciais Edson Aparecido Ciência da Computação – Faculdades de Valinhos, São Paulo, Brasil. Disponível em:>https://www.aedb.br/seget/arquivos/artigos05/318_Artigo-002.pdf (Artigo).

Gil, A. C. Como elaborar projetos de pesquisa. (2010) São Paulo: atlas. (Livro).

Microfocus. (2020) White Paper Conectividade e modernização de aplicativos. COBOL aos 60: uma lenda viva celebrando o valor duradouro do COBOL. Disponível em>[https://3-Cobol-ao-60\(1\).pdf](https://3-Cobol-ao-60(1).pdf). (Livro).

Sads. (2018). BRB – Banco de Brasília sai vencedor no Prêmio efinance com projetos conduzidos juntamente com a Spread. Disponível em> <https://spread.com.br/tag/sads/> (Blog de Cases).

Spread Tecnologia (2014). Migração e Modernização de ambientes Mainframes. Disponível em <https://marketingspread.wixsite.com/spread/sads>. (Site).

Tigre, P. B; Noronha, V. B. (2013). Do mainframe à nuvem: inovações, estrutura industrial e modelos de negócios nas tecnologias da informação e da comunicação, Revista de Administração USP, 48(1), 114-127. DOI: <https://doi.org/10.5700/rausp1077> (Artigo de Revista)

Support System for Systems Development

ABSTRACT

This work sought to analyze the SADS tool (Systems Development Support System) demonstrating the advantages of its use in systems development. The case analysis was carried out at the company Spread Tecnologia, which holds the rights to the respective tool, and which granted authorization to produce this work in its organizational environment for academic purposes. From this perspective, with the challenge of presenting a solution that optimizes access, manipulation and integration of data with support from the COBOL language, the company MSA-INFOR Sistemas e Automação Ltda produced this tool aiming to contribute significantly to the efficiency and success of development projects. Systems, which is used to assist in the development and maintenance of systems, especially on the Unisys platform. Thus, the case study presented provides an in-depth understanding of the respective technology in systems development, in addition to stimulating innovation and essential skills for information technology professionals who contribute to solving complex problems. The results found highlighted the effectiveness of SADS in efficient data management, advanced systems modeling and overcoming challenges in the development process. In this way, it is understood that the proposal presented by the professionals who authored this study are suitable for optimizing processes that involve systems, and promoting effective data management, reinforcing the relevance of SADS in the academic and professional scenario.

Keywords: Ferramenta SADS; Language Cobol; Systems.

Sistema de Apoyo al Desarrollo de Sistemas

ABSTRACTO

Este trabajo buscó analizar la herramienta SADS (Systems Development Support System) demostrando las ventajas de su uso en el desarrollo de sistemas. El análisis del caso se realizó en la empresa Spread Tecnologia, propietaria de los derechos de la respectiva herramienta, y que otorgó autorización para producir este trabajo en su entorno organizacional con fines académicos. Desde esta perspectiva, con el desafío de presentar una solución que optimice el acceso, manipulación e integración de datos con soporte del lenguaje COBOL, la empresa MSA-INFOR Sistemas e Automação Ltda produjo esta herramienta con el objetivo de contribuir significativamente para la eficiencia y el éxito de proyectos de desarrollo. Systems, que se utiliza para ayudar en el desarrollo y mantenimiento de sistemas, especialmente en la plataforma Unisys. Así, el estudio de caso presentado proporciona una comprensión profunda de la tecnología respectiva en el desarrollo de sistemas, además de estimular la innovación y habilidades esenciales para los profesionales de tecnologías de la información que contribuyen a la solución de problemas complejos. Los resultados encontrados resaltaron la efectividad de SADS en la gestión eficiente de datos, el modelado de sistemas avanzados y la superación de desafíos en el proceso de desarrollo. De esta manera, se entiende que las propuestas presentadas por los profesionales autores de este estudio son adecuadas para optimizar procesos que involucran sistemas y promover una gestión eficaz de datos, reforzando la relevancia de SADS en el escenario académico y profesional.

Palabras clave: Ferramenta SADS; Idioma Cobol; Sistemas.