|  |  |
| --- | --- |
|  | 🞂**Documentación Técnica Sistema: $sistema$**  **Componente CBL - $componente$** |
|  |  |
|  | **Powered by: SysViewSoft, S. A. de C. V.** 🞂 **Registrar**🞂 **27/04/2016** |
|  |  |
|  | Objetivo  El objetivo de este documento es proporcionar información del componente. |
|  | Propiedades del Componente  Tipo: Es un componente tipo COBOL  Objetivo:  Descripción: $descripción$    Autor:  Fecha:  Historial de Versiones. Última versión |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

<SVSOUTPUT>Documentación Técnica Componente|$ruta$$sistema$\COMPONENTES\$clase$\ DT\_$sistema$\_$clase$\_$biblioteca$ \_$componente$.docx

Documentación Técnica Sistema: $sistema$

Componente CBL - $componente$

## Alcance

Se han integrado al presente proceso de Minería de Aplicaciones piezas incluyendo versiones; con este inventario de componentes el presente documento incluye los siguientes productos:

* Programa fuente
* Diagrama de Análisis de Impacto
* Diagrama de Componente
* Diagrama de Flujo
* Diagrama Jerárquico
* Lista de Componentes
* Lista de Dependencias de Componentes
* Referencias cruzadas
* Diagrama de bloques
* Lista Jerárquica de Componentes

## Algoritmos en el Componente

Esta sección incluye la lista de cálculos asociados al componente.

Código fuente

Esta sección permite conocer el código del componente.

|  |
| --- |
| **<SVSICON>Ver Fuente|$ruta$$sistema$\$clase$\fFuente $clase$ $biblioteca$ $componente$.txt** |

|  |
| --- |
| **<SVSWORD>Ver Fuente|$ruta$$sistema$\$clase$\fFuente $clase$ $biblioteca$ $componente$.txt** |

Diagrama de Flujo Procedural

El diagrama procedural muestra a partir de un componente el Flujo que presenta su programación.

|  |
| --- |
| **<SVSICON>Diagrama de Flujo |$ruta$$sistema$\$clase$\fDgrFlujo $clase$ $biblioteca$ $componente$.pdf** |

|  |
| --- |
| **<SVSWORD>Diagrama de Flujo |$ruta$$sistema$\$clase$\fDgrFlujo $clase$ $biblioteca$ $componente$.pdf** |

Diagrama Jerárquico

El Diagrama Jerárquico presenta de forma vertical los llamados que se hacen de las rutinas internas en un programa Cobol entre sí.

El diagrama parte de la rutina principal del programa y va mostrando desde donde son llamadas el resto de las rutinas, regularmente incluye comandos Perform en sus distintos formatos.

|  |
| --- |
| **<SVSICON>Diagrama Jerárquico|$ruta$$sistema$\$clase$\fDgrJerarquico $clase$ $biblioteca$ $componente$.pdf** |

|  |
| --- |
| **<SVSWORD>Diagrama Jerárquico|$ruta$$sistema$\$clase$\fDgrJerarquico $clase$ $biblioteca$ $componente$.pdf** |

Diagrama de Bloque

El diagrama de bloque tradicional permite visualizar de manera muy sencilla los componentes de INPUT y los componentes de OUTPUT a partir de un programa Cobol. El diagrama presenta también el modo en que se utiliza el componente en el programa.

|  |
| --- |
| **<SVSICON>Diagrama de Bloque|$ruta$$sistema$\$clase$\fDgrBloques $clase$ $biblioteca$ $componente$.pdf** |

|  |
| --- |
| **<SVSWORD>Diagrama de Bloque|$ruta$$sistema$\$clase$\fDgrBloques $clase$ $biblioteca$ $componente$.pdf** |

Lista de Componentes

Utilice la lista de componentes para conocer los componentes que son referenciados en este componente.

La Lista de Componentes muestra un proceso Drill Down a partir del componente.

Realice filtros sobre la lista para atender el tipo específico de componentes de su interés.

|  |
| --- |
| <SVSICON>Lista de Componentes|$ruta$$sistema$\$clase$\fLstComponentes $clase$ $biblioteca$ $componente$.csv |

|  |
| --- |
| <SVSWORD>Lista de Componentes|$ruta$$sistema$\$clase$\fLstComponentes $clase$ $biblioteca$ $componente$.csv |

Lista de Dependencia de Componentes

La Lista de Dependencia de Componentes es similar a la Lista de Componentes, pero a diferencia de esta, la Lista de Dependencia va marcando los Niveles Jerárquicos (NIVEL) por número. Igualmente es posible obtener esta lista en formato de Excel.

Las características de cada tipo de componente son listadas, por ejemplo, cuando se trata de un archivo indexado (tradicional), se establece el modo en que se utiliza.

|  |
| --- |
| **<SVSICON>Lista de Dependencia de Componentes|$ruta$$sistema$\$clase$\fLstDependencias $clase$ $biblioteca$ $componente$.csv** |

|  |
| --- |
| **<SVSWORD>Lista de Dependencia de Componentes|$ruta$$sistema$\$clase$\fLstDependencias $clase$ $biblioteca$ $componente$.csv** |

Diagrama de Análisis de Impacto

Es la representación gráfica que se obtiene como resultado de realizar un recorrido ascendente del árbol creado en la base de conocimiento para un determinado sistema. El Análisis de Impacto se genera a partir de un determinado tipo de componente.

|  |
| --- |
| **<SVSICON>Análisis de Impacto|$ruta$$sistema$\$clase$\fDgrAImpacto $clase$ $biblioteca$ $componente$.pdf** |

|  |
| --- |
| **<SVSWORD>Análisis de Impacto|$ruta$$sistema$\$clase$\fDgrAImpacto $clase$ $biblioteca$ $componente$.pdf** |

Diagrama de Componentes

El diagrama de componentes muestra el árbol (Drill Down) de los componentes que son usados o llamados desde el componente de mayor jerarquía, similar al proceso Drill Down o Lista de Componentes.

|  |
| --- |
| **<SVSICON>Diagrama de Componentes|$ruta$$sistema$\$clase$\fDgrProcesos $clase$ $biblioteca$ $componente$.pdf** |

|  |
| --- |
| **<SVSWORD>Diagrama de Componentes|$ruta$$sistema$\$clase$\fDgrProcesos $clase$ $biblioteca$ $componente$.pdf** |

Referencias Cruzadas

El proceso de Referencias Cruzadas permite establecer a partir de un componente Padre, la relación que tienen sus componentes (Dependientes) con respecto del resto de componentes que integran un sistema. Esto es:

Se presentan en función a Renglones y Columnas los componentes que tienen un uso común.

Renglón: Integrado por los componentes de “nivel inferior” de acuerdo con el “Árbol” en un recorrido “Descendente”.

Columna: Componentes que hacen uso del componente Renglón.

Las referencias cruzadas deben partir de un componente “Base” dentro del “Árbol” construido sobre la base de conocimiento.

En la práctica este informe se utiliza para visualizar “donde” se están compartiendo los componentes que usa determinado componente Base.

|  |
| --- |
| **<SVSICON>Referencias Cruzadas|$ruta$$sistema$\$clase$\fLstRefCruzadas $clase$ $biblioteca$ $componente$.csv** |

|  |
| --- |
| **<SVSWORD>Referencias Cruzadas|$ruta$$sistema$\$clase$\fLstRefCruzadas $clase$ $biblioteca$ $componente$.csv** |

Código Muerto

Tiene como objetivo mostrar la parte del código fuente que se ejecuta y consume tiempo de cómputo en algo que jamás se utiliza.

Uso de Rutinas

|  |
| --- |
| **<SVSICON>Código Muerto Uso Rutinas|$ruta$$sistema$\$clase$\fCodigoMuerto1 $clase$ $biblioteca$ $componente$.csv** |

|  |
| --- |
| **<SVSWORD>Código Muerto Uso Rutinas|$ruta$$sistema$\$clase$\fCodigoMuerto1 $clase$ $biblioteca$ $componente$.csv** |

Código Muerto

(Continúa…

Tiene como objetivo mostrar la parte del código fuente que se ejecuta y consume tiempo de cómputo en algo que jamás se utiliza.

Uso de Variables

|  |
| --- |
| **<SVSICON>Código Muerto Uso Variables|$ruta$$sistema$\$clase$\fCodigoMuerto2 $clase$ $biblioteca$ $componente$.csv** |

|  |
| --- |
| **<SVSWORD>Código Muerto Uso Variables|$ruta$$sistema$\$clase$\fCodigoMuerto2 $clase$ $biblioteca$ $componente$.csv** |

Código Muerto

(Continúa…

Tiene como objetivo mostrar la parte del código fuente que se ejecuta y consume tiempo de cómputo en algo que jamás se utiliza.

Código Eliminado

|  |
| --- |
| **<SVSICON>Código Muerto Código Eliminado|$ruta$$sistema$\$clase$\fCodigoMuerto3 $clase$ $biblioteca$ $componente$.csv** |

|  |
| --- |
| **<SVSWORD>Código Muerto Código Eliminado|$ruta$$sistema$\$clase$\fCodigoMuerto3 $clase$ $biblioteca$ $componente$.csv** |

Código Muerto

(Continúa…

Tiene como objetivo mostrar la parte del código fuente que se ejecuta y consume tiempo de cómputo en algo que jamás se utiliza.

Líneas Modificadas

|  |
| --- |
| **<SVSICON>Código Muerto Líneas Modificadas|$ruta$$sistema$\$clase$\fCodigoMuerto4 $clase$ $biblioteca$ $componente$.csv** |

|  |
| --- |
| **<SVSWORD>Código Muerto Líneas Modificadas|$ruta$$sistema$\$clase$\fCodigoMuerto4 $clase$ $biblioteca$ $componente$.csv** |

Complejidad de Componentes

Conocer que tan compleja o sencilla resulta la actividad de mantenimiento para un programa o bien, conocer la complejidad implicada en migrar / convertir un programa hacia otra plataforma se resuelve a través de calcular la complejidad de un programa. Existen diferentes algoritmos en el mercado perfectamente documentados y que se utilizan en muchas organizaciones para tales efectos. Algunos de esos algoritmos son:

Complejidad de Halstead.

Se apoya en métricas basadas en el cálculo de palabras reservada y operadores lógicos (Operadores) así como también en variables, constantes numéricas e identificadores (Operandos), escritos en el código de un programa cobol. Las métricas permiten calcular los siguientes conceptos:

* Vocabulario

Es el resultado de sumar palabras reservadas, operadores lógicos, variables y constantes todas ellas únicas en el código fuente, es decir, no se acumulan cuando éstas se repiten en su uso.

* Longitud

Es la medida del tamaño de un programa en relación al total de palabras reservadas, operadores lógicos y variables usados. A mayor tamaño, mayor esfuerzo para comprenderlo y mantenerlo.

* Duración

Es el resultado de aplicar el cálculo y obtener la suma del logaritmo de palabras reservadas y operadores lógicos más el logaritmo de variables y constantes.

* Volumen

Da un peso extra de acuerdo al número de palabras reservadas, operadores lógicos, variables, constantes, etc. Todos ellos únicos y que por ese número lo hagan más difícil de entender y mantener.

* Dificultad

Es la medida de la complejidad que tiene un programa para escribirlo o entenderlo, por ejemplo, cuando se hace la revisión de código.

* Esfuerzo

Ofrece una medida del trabajo requerido para desarrollar un nuevo programa. Desde el punto de vista del mantenimiento, el esfuerzo se puede interpretar como una medida del trabajo requerido para comprender un software ya desarrollado. Aumenta proporcionalmente con el volumen

Complejidad de Componentes

(Continúa…

* Tiempo para Entendimiento del Programa

Es el resultado de calcular el Esfuerzo entre la unidad de medida de 18 segundos.

* Número de Errores de Entrega

Es el resultado de calcular el Volumen entre un factor establecido.

Detalle de Complejidad Halstead en Cobol por Componente

|  |
| --- |
| **<SVSICON> Detalle Complejidad Halstead Componente Cobol|$ruta$$sistema$\$clase$\fComHalsteadDet $clase$ $biblioteca$ $componente$.xlsx** |

|  |
| --- |
| **SVSWORD> Detalle Complejidad Halstead Componente Cobol|$ruta$$sistema$\$clase$\fComHalsteadDet $clase$ $biblioteca$ $componente$.xlsx** |

Complejidad de Componentes

(Continúa…

Totales de Complejidad Halstead en Cobol por Componente

|  |
| --- |
| **<SVSICON>Totales Complejidad Halstead Componente Cobol|$ruta$$sistema$\$clase$\fComHalsteadTot $clase$ $biblioteca$ $componente$.xlsx** |

|  |
| --- |
| **<SVSWORD>Totales Complejidad Halstead Componente Cobol|$ruta$$sistema$\$clase$\fComHalsteadTot $clase$ $biblioteca$ $componente$.xlsx** |

Complejidad de Componentes

(Continúa…

Complejidad de Mc Cabe.

Permite establecer en el código de un programa cobol el nivel de mantenibilidad o esfuerzo necesario para poder probar todos sus caminos, ya que cuanto más compleja sea la lógica de un código, más difícil será de entender, mantener y probar.

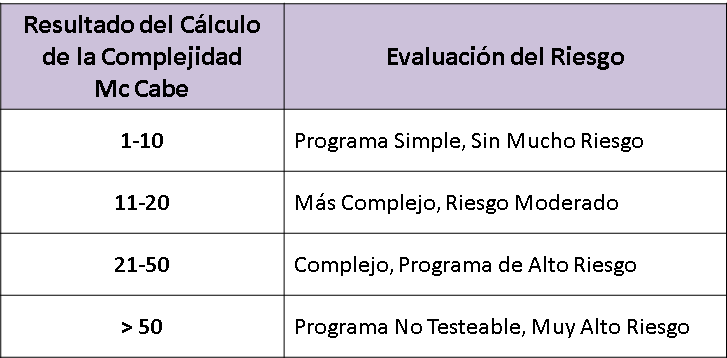
Para realizar el cálculo de la complejidad Mc Cabe se utilizan los siguientes conceptos:

* Número de aristas.

Establece el número de caminos que puede tener la lógica de un programa. En instrucciones relativas a condiciones se puede tener más de un camino a seguir en la lógica mientras que en las instrucciones que se realizan una después de la otra, es solamente un camino.

* Número de nodos o sentencias del programa.

Los conceptos anteriores permiten establecer la evaluación de riesgo hacia el código una vez que se realizó el cálculo de la Complejidad Mc Cabe:



Complejidad de Componentes

(Continúa…

Detalle de Complejidad Mc Cabe en Cobol por Componente

|  |
| --- |
| **<SVSICON>Detalle Complejidad Mc Cabe Componente Cobol|$ruta$$sistema$\$clase$\fComMcCabeDet $clase$ $biblioteca$ $componente$.xlsx** |

|  |
| --- |
| **<SVSIWORD>Detalle Complejidad Mc Cabe Componente Cobol|$ruta$$sistema$\$clase$\fComMcCabeDet $clase$ $biblioteca$ $componente$.xlsx** |

Complejidad de Componentes

(Continúa…

Totales de Complejidad Mc Cabe en Cobol por Componente

|  |
| --- |
| **<SVSICON>Totales Complejidad Mc Cabe Componente Cobol|$ruta$$sistema$\$clase$\fComMcCabeTot $clase$ $biblioteca$ $componente$.xlsx** |

|  |
| --- |
| **<SVSWORD>Totales Complejidad Mc Cabe Componente Cobol|$ruta$$sistema$\$clase$\fComMcCabeTot $clase$ $biblioteca$ $componente$.xlsx** |

Lista Jerárquica de Componentes

En esta sección podrá encontrar una lista organizada por distintos niveles de componentes iniciando en el nivel más alto el componente objetivo de este documento y en los niveles subsecuentes aquellos componentes de los que hace uso.

Es de gran utilidad esta información ya que de cada uno de esos componentes podrá encontrar su respectiva documentación.

**<SVSTABDEP>**