**MANUAL DE LOS SERVICIOS**

**ACMR**

# INFORMACION DEL DOCUMENTO

## Objetivo

El siguiente documento tiene por objetivo presentar los conceptos, funciones técnicas y operativas así como también el estado actual de situación de los servicios del sistema ACMR

## Control de Versiones

El presente documento ha sido sometido a las siguientes revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Autor** | **Comentarios** |
|  |  |  |  |
| Borrador 1 | 10.02.2017 | Pedro Curich | Versión inicial del documento |
| Revisión I | 28.02.2017 | Pedro Curich | Actualización |
| Revisión II | 13.06.2019 | Pedro Curich | Actualización por el cambio requerido |
|  |  |  |  |
|  |  |  |  |

## Aprobadores

El presente documento requiere de la aprobación de:

|  |  |
| --- | --- |
| **Nombre y Apellido** | **Función** |
|  |  |
| Brayan Chávez | Analista Funcional |
| Pedro Curich | Desarrollador y Arquitecto |
| Contador |  |
| Coronel |  |
| Ronie | Líder GD |

# Propósitos de Negocio

## Resumen de funcionalidad de negocios

Es un sistema de servicios que permiten exportar e importar archivos de diferente fuente de información tales como COPERE o CPMP

# Proceso de negocios / Funciones Cubiertas.

* Exportar datos para el COPERE
* Exportar datos para la CPMP
* Importar la respuesta desde el COPERE
* Importar la respuesta desde la CPMP
* Sincronizar y actualizar los pagos en el sistema

## Información sensible almacenada en la aplicación:

Lista de asociados registrados en la aplicación, montos aportados por cada personal militar en actividad o en retiro y los préstamos con deudas que poseen dicho personal militar

## Identificación de los Procesos Críticos

* **Aportaciones programadas de asociados en actividad o retiro**

**Periodos**: Se realizada una vez al mes en una fecha determinada y establecida por el usuario desde el módulo Configuración -> Programar Servicios  
**Pago de la aportación (Cabecera)**: registra lo aportado por cada asociado  
**Pago en una fecha determinada (detalle)**: monto correspondiente al pago, este tiene un estado que identifica si ha sido pagado o no en base a una fecha de programación que existe en el sistema  
**Estados (StateId):** Pendiente = 1, EnProceso = 2, PagoParcial = 3, Pagado = 4, SinLiquidez = 5, Devolucion = 6, PagoPersonal = 7

* **Cobros programados del Apoyo Social Económico para los asociados en actividad o retiro  
  Periodos**: Se realizada una vez al mes en una fecha determinada y establecida por el usuario desde el módulo Configuración -> Programar Servicios  
  **Pago del Apoyo (Cabecera)**: registra los datos básicos del Apoyo, tales como N° de cuotas, monto del préstamo, etc.

**Pago en una fecha determinada (detalle)**: monto correspondiente al cobro de la Cuota calendarizada  
**Estados (StateId):** Pendiente = 1, EnProceso = 2, PagoParcial = 3, Pagado = 4, SinLiquidez = 5, Anulado = 6, PagoPersonal = 7, Devolucion=8, Cancelado=9

* **Sincronizador de los pagos realizados  
  Periodos**: Se ejecuta cada vez que se ingresa un archivo de respuesta, ya sea del COPERE o de CPMP  
  **Sin Liquidez**: Ocurre cuando en la respuesta no se encuentra un asociado que si se llegó a enviar  
  **Pago parcial**: Ocurre cuando en la respuesta el monto que se informa no es igual (menor) al monto que se envió para el cobro (Aportación + Apoyo)  
  **Pago total**: Ocurre cuando en la respuesta el monto que se informa es igual al monto que se envió para el cobro (Aportación + Apoyo) y logra cubrir la cuota

# ÁREAS USUARIAS Y SOPORTE TECNICO

## Usuarios del Aplicativo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N° | Área | Nombres y Apellidos | Cargo | Teléfono | Correo electrónico |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

## Equipo de soporte del aplicativo

|  |  |  |  |
| --- | --- | --- | --- |
| N° | Nombres y Apellidos | Teléfono | Correo electrónico |
| 1 | Pedro Curich Gonzales | 973 905 013 | [curichpedro@gmail.com](mailto:curichpedro@gmail.com) |
| 2 | Brayan Jesús Chávez Olortegui | 959 034 989 | [jbcho8@gmail.com](mailto:jbcho8@gmail.com) |

# ARQUITECTURA TECNOLÓGICA DE LA APLICACIÓN

## Datos Generales

|  |  |
| --- | --- |
| **Nombre de la Aplicación** | Servicios ACMR |
| **Versión actual en producción** | 1.6 |
| **Principal tecnología involucrada** | Microsoft .NET 4.5 , SQL Server 2008 R2 |

## Documentación técnica

Manual para el entendimiento del proceso realizado por los servicios

## Resumen de Ambiente Técnico

Servicios Windows con tecnología .NET 4.5 y base de datos SQL Server 2008 R2 montado sobre el sistema de servicios de Windows

## Arquitectura de la Infraestructura

## 

|  |
| --- |
| C:\Users\Pedro Curich\Downloads\SERVICIOS ACMR (1).png |
| Imagen1 |

## Arquitectura de los servicios de exportación (COPERE, CPMP)

|  |
| --- |
| C:\Users\Pedro Curich\Downloads\Untitled Diagram (2).png |
| Imagen2 |

## Arquitectura de los servicios de carga (COPERE, CPMP)

|  |
| --- |
| C:\Users\Pedro Curich\Downloads\Untitled Diagram (3).png |
| Imagen3 |

## Herramientas requeridas en el servidor de desarrollo

|  |  |
| --- | --- |
| **Sistema Operativo** | Windows 8, Windows 10 |
| **Lenguaje de Programación** | .NET 4.5 |
| **Servidor de Aplicaciones** | IIS 10 |
| **Base de Datos** | SQL SERVER R2 |
| **Otros** | Funciona en los tres navegadores (Explorer, Firefox, Chrome) |

## Herramientas requeridas en la estación de trabajo

|  |  |
| --- | --- |
| **Sistema Operativo** | Windows 8, Windows 10 |
| **.NET Framework** | .NET 4.5 |
| **JAVA virtual machine** | N.A |
| **Cliente Oracle** | N.A |
| **MS SQL Server** | SQL SERVER R2 |
| **Cliente DB2** | N.A |
| **Navegador web** | Explorer, Firefox, Chrome |
| **Paquete de Office** | N.A |
| **Otros** | N.A |

## Directorios utilizados por la aplicación

C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Caja.In\Deploy  
C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Caja.Out\Deploy  
C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Copere.In\Deploy  
C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Copere.Out\Deploy

Bachero BindDirectory.bat proporcionado

## Requisitos para la instalación

* DLL
* .NET 4.5
* Quartz.dll
* Topshelf.dll

# ARQUITECTURA DE DATOS

## Tablas Maestras

|  |
| --- |
|  |
| Imagen4 |

## Tablas Aportaciones

|  |
| --- |
|  |
| Imagen5 |

## Tablas Apoyo Social Económico

|  |
| --- |
|  |
| Imagen6 |

# ARQUITECTURA FUNCIONAL DE LA APLICACIÓN

## Documentación funcional

Diagrama de clases – info

|  |
| --- |
|  |
| Imagen7 |

Diagrama de clases – report

|  |
| --- |
|  |
| Imagen8 |

## Diagrama de flujo de la Aplicación

Diagrama de secuencia:

|  |  |
| --- | --- |
| Exportar | C:\Users\Pedro Curich\Downloads\Untitled Diagram (4).png |
| Importar |  |
| Sincronizador |  |

# ESPECIFICACIÓN DE LAS FUNCIONALIDADES DEL SISTEMA

## Servicio Ks Batch Caja In

|  |  |
| --- | --- |
| **Servicio:** | P001 Servicio de Importación para Caja Privada Militar Policial |
| **Descripción:** | Servicio que permite importar los cobros realizados desde la caja privada militar policial (CPMP) hacia el sistema ACMR |
| **Periodo:** | El Bachero se ejecuta cada vez que se ingresa un archivo en la ruta de lectura con el siguiente nombre : PreCajaWakeUp.txt o CajaWakeUp.txt |
| **Directorio Lectura** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Caja.In\Read |
| **Directorio Escritura** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Caja.In\Process |
| **Directorio Log** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Caja.In\Log |
| **Tipo:** | FileSystemWatcher |
| **Subtipo:** | FileCreated |
| **Pasos** | 1. Se deja el archivo de respuesta con el formato CPMP YYYY MM\_6008.txt 2. El proceso detecta que existe un archivo por procesar, este duerme 3 segundos para asegurar que sea atómico el copiado 3. Llena las siguientes variables del archivo APP.config:   Path: Directorio  Connection:Base de datos  SysName: Nombre del servicio 4. Obtiene el último registro de la base de datos que se ejecuto 5. Del nombre del archivo ingresado se extrae los siguientes valores: 6. Se procede a leer línea a línea la información del archivo obteniendo la siguiente información: 7. Una vez leído las líneas del archivo y transformado a un objeto, se procede a buscar el registro de la tabla **Report** el cual indica por **source** y por **periodo** la información que se va a actualizar, cambia el estado de 1 a 2 8. Inserta la información leída para su posterior procesamiento 9. Actualiza el registro de la tabla **ScheduleBatch** el cual prepara la información para la siguiente lectura |

## Servicio Ks Batch Copere In

|  |  |
| --- | --- |
| **Servicio:** | P002 Servicio de Importación para COPERE |
| **Descripción:** | Servicio que permite importar los cobros realizados desde la planilla de los oficinales en actividad (COPERE) |
| **Periodo:** | El Bachero se ejecuta cada vez que se ingresa un archivo en la ruta de lectura con el siguiente nombre : PreCopereWakeUp.txt o CopereWakeUp.txt |
| **Directorio Lectura** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Copere.In\Read |
| **Directorio Escritura** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch. Copere.In\Process |
| **Directorio Log** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch. Copere.In\Log |
| **Tipo:** | FileSystemWatcher |
| **Subtipo:** | FileCreated |
| **Pasos** | 1. Se deja el archivo de respuesta con el formato EC.ACT.NOMINAL\_8001\_MMYYYY.txt 2. El proceso detecta que existe un archivo por procesar, este duerme 3 segundos para asegurar que sea atómico el copiado 3. Llena las siguientes variables del archivo APP.config:   Path: Directorio  Connection:Base de datos  SysName: Nombre del servicio 4. Obtiene el último registro de la base de datos que se ejecuto 5. Del nombre del archivo ingresado se extrae los siguientes valores: 6. Se procede a leer línea a línea la información del archivo obteniendo la siguiente información: 7. Una vez leído las líneas del archivo y transformado a un objeto, se procede a buscar el registro de la tabla **Report** el cual indica por source y por periodo la información que se va a actualizar, cambia el estado de 1 a 2 8. Inserta la información leída para su posterior procesamiento 9. Actualiza el registro de la tabla **ScheduleBatch** el cual prepara la información para la siguiente lectura |

## Servicio Ks Batch Caja Out

|  |  |
| --- | --- |
| **Servicio:** | P003 Servicio de Exportación para CPMP |
| **Descripción:** | Servicio que permite exportar los cobros por realizar hacia la CPMP desde el sistema ACMR |
| **Periodo:** | El Bachero se ejecuta 30 segundos |
| **Directorio Lectura** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Caja.Out\Read |
| **Directorio Escritura** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Caja.Out\Process |
| **Directorio Log** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Caja.Out\Log |
| **Tipo:** | Quartz |
| **Subtipo:** | Ijob |
| **Pasos** | 1. Se deja el archivo de respuesta con el formato CPMP YYYY MM\_6008 MESLETRAS.txt 2. Llena las siguientes variables del archivo APP.config:   Path: Directorio  Connection:Base de datos  SysName: Nombre del servicio 3. Obtiene el último registro de la base de datos que se ejecuto      1. Se busca el servicio Ks.Batch.Caja.Out y se obtiene la última ejecución realizada 2. Si el servicio está activo para ser ejecutado se procede a validar si físicamente existe el archivo de salida, de lo contrario se termina el proceso 3. La validación física se da con la existencia del archivo    con los valores extraídos del batch a ejecutarse, es decir los valores que están en la base de datos los cuales van a ser procesados 4. Se procede a extraer los asociados hacia el archivo de envió al copere 5. Se obtienen los Ids de los asociados que se encuentran en **COPERE**  mediante la siguiente consulta: 6. Se obtienen las aportaciones activas de los asociados obtenidos en el punto 8 mediante la siguiente consulta    El estado de la aportación debe ser activa () y el estado del detalle es 1:    Esto para el mes y año en busqueda 7. Se extrae los apoyos sociales económicos activos de los asociados obtenidos en el punto 8 mediante la siguiente consulta y cuya cuota este para el periodo de consulta. Esto mediante la siguiente consulta:    De igual modo con el estado de activo para el apoyo social () y para la cuota que se encuentre en estado pendiente con fecha actual del bachero 8. Cabe resaltar que de esta forma se independiza la extracción de la información teniendo los aportes y el apoyo social económico separado en base a los aportantes que son de COPERE 9. El siguiente paso es filtrar aquellos asociados que poseen una aportación vigente para cobrar o algún pago pendiente de una cuota de un apoyo social económico    En donde **FileOut** es un diccionario de 3 valores : key del asociado, key del asociado con la linea a ingresar al archivo; ademas **ReportOut** contiene el objeto a serializar para la creacion de la linea en el archivo (Info) 10. Se sumariza las aportaciones con el apoyo social económico para crear el total que se enviara a **COPERE** 11. Finalmente se elimina de la tabla **report** los registros que hace referencia a caja Out Y Caja In del años en proceso para luego crear nuevamente la dupla 12. Se crea el archivo físico con información a ser enviada a COPERE 13. Se actualiza el bachero con la ejecución realizada y queda esperando la siguiente ejecución con fecha mes +1 del actual 14. Actualiza el registro de la tabla **ScheduleBatch** |

## Servicio Ks Batch Copere Out

|  |  |
| --- | --- |
| **Servicio:** | P004 Servicio de Exportación para COPERE |
| **Descripción:** | Servicio que permite exportar los cobros por realizar hacia el COPERE desde el sistema ACMR |
| **Periodo:** | El Bachero se ejecuta 30 segundos |
| **Directorio Lectura** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Copere.Out\Read |
| **Directorio Escritura** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Copere.Out\Process |
| **Directorio Log** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Copere.Out\Log |
| **Tipo:** | Quartz |
| **Subtipo:** | Ijob |
| **Pasos** | 1. Se deja el archivo de respuesta con el formato 8001\_YYYYMM00.txt 2. Llena las siguientes variables del archivo APP.config:   Path: Directorio  Connection:Base de datos  SysName: Nombre del servicio 3. Obtiene el último registro de la base de datos que se ejecuto 4. Se busca el servicio Ks.Batch.Copere.Out y se obtiene la ultima ejecucion realizada 5. Si el servicio esta activo para ser ejecutado se procede a validar si fisicamente existe el archivo de salida, de lo contrario se termina el proceso 6. La validacion fisica se da con la existecia del archivo     con los valores extraidos del batch a ejecutarse, es decir los valores que estan en la base de datos son los que se van a procesar 7. Se procede a extraer los asociados hacia el archivo de envio al Copere 8. Se optienen los Ids de los asociados que se encuentran en **COPERE**  mediante la siguiente consulta: 9. Se obtienen las aportaciones activas de los asociados obtenidos en el punto 8 mediante la siguiente consulta    El estado de la aportacion debe ser activa () y el estado del detalle es 1:    Esto para el mes y año en busquedad 10. Se extrae los apoyo social economicos activos de los asociados obtenidos en el punto 8 mediante la siguiente consulta y cuya cuota este para el periodo de consulta. Esto mediante la siguiente consulta:    De igual modo con el estado de activo para el apoyo social () y para la couta que se encuentre en estado pendiente cn con fecha actual del bachero 11. Cabe resaltar que de esta forma se independiza la extraccion de la informacion teniendo los aportes y el apoyo social economico separado en base a los aportantes que son de COPERE 12. El siguiente paso es filtrar aquellos asociados que poseen una aportacion vigente para cobrar o algun pago pendiente de una couta de un apoyo social economico    En donde **FileOut** es un diccionario de 3 valores : key del asociado, key del asociado con la linea a ingresar al archivo; ademas **ReportOut** contiene el objeto a serializar para la creacion de la linea en el archivo (Info) 13. Las aportaciones y los apoyos social se envian por separadoen la interfaz al **COPERE** 14. Finalmente se elimina de la tabla **report** los registros que hace referencia a Copere Out Y Copere In del años en proceso para luego crear nuevamente la dupla 15. Se crea el archivo fisico con informacion a ser enviada a COPERE 16. Se actualiza el bachero con la ejecucion realizada y queda esperando la siguiente ejecucion con fecha mes +1 del actual 17. Actualiza el registro de la tabla **ScheduleBatch** |

## Servicio Ks Batch Merge

|  |  |
| --- | --- |
| **Servicio:** | P005 Servicio de Sincronizacion tabto para COPERE como para CPMP |
| **Descripción:** | Servicio que permite sincronizar la información de los archivos de entrada como los de salida, ya sea para COPERE como para CPMP |
| **Periodo:** | El Bachero se ejecuta cada vez que termina el servicio **Ks.Batch.Copere.In** o **Ks.Batch.Caja.In** |
| **Directorio Lectura** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch.Merge \Read |
| **Directorio Escritura** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch. Merge \Process |
| **Directorio Log** | C:\inetpub\wwwroot\Acmr\App\_Data\Service\Ks.Batch. Merge \Log |
| **Tipo:** | FileSystemWatcher |
| **Subtipo:** | FileCreated |
| **Pasos** | 1. Se ejecuta cuando el servicio **Ks.Batch.Copere.In o Ks.Batch.Caja.In** terminan de cargar la respuesta (CPMP o COPERE) en la base de datos 2. Llena las siguientes variables del archivo APP.config:   Connection:Base de datos 3. Obtiene el último registro de la base de datos que se ejecuto 4. Se busca la informacion de la tabla **report** que contiene el par de registros (StateId=1 StateId=2) con el mismo ParentKey y se extrae 5. La data que se extrae contiene un archivo xml con la data cargada tanto de los procesos de salida como los de entrada (IN y OUT ) 6. Hasta este momento se cuenta con 2 listas de tipo List<Info>. Luego se divide en 4 listas      con la finalidad de poder identificar la informacion que posee cada lista  ademas del origen de los datos 7. Si se tiene la dupla entre caja o copere se procede con la extraccion de data 8. Ya sea para caja o para copere se cuenta con 6 listas, 3 para aportaciones y 3 para apoyo social economico 9. Para Aportaciones se tiene 3 listas que separan la informacion bajo 3 criterios: Sin liquidez, pagos completos y pagos parciales 10. Para el Apoyo social eonomato se tiene 3 listas que separan la informacion bajo 3 criterios: Sin liquidez, pagos completos y pagos parciales 11. Se inicia con la identificacoin de de los estados de los registros pudiendo ser de 4 tipos:  \* Sin liquidez en aportaciones y Apoyo  \* Pago completo en aportaciones y sin liquidez en Apoyo  \* Pago por partes en apotaciones y sin liquidez en Apoyo  \* Pago total en apotaciones y en partes en Apoyo 12. Se separa la informacion relacionada a aportaciones y a apoyo social 13. Se procede a actualizar en Base de datos la informacion de los saldos mediante los siguientes StoreProcedure:  UpdateContributionPaymentCopere UpdateContributionPaymentCaja UpdateLoanPaymentCopere UpdateLoanPaymentCaja 14. Para el caso de UpdateContributionPayment\* se crean 3 tablas temporales en donde se colocan los que son de pago parcial (3), pagado (4) y sin liquidez (5). Luego se actualiza la tabla **ContributionPayment** con el resultado optenido de cada estado.  Se ejecuta el StoreProcedure **FixQuota\*** el cual regulariza las coutas futuras que se realizan 15. Para el caso de UpdateLoanPayment\*  se crean 3 tablas temporales en donde se colocan los que son de pago parcial (3), pagado (4) y sin liquidez (5). Luego se actualiza la tabla **LoanPayment** con el resultado optenido de cada estado |