



FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA PROFESIONAL DE INGENIERÍA DE COMPUTACIÓN Y SISTEMAS

**DISEÑO DE LA INFRAESTRUCTURA GRID PARA LA  
OPTIMIZACIÓN A LA EJECUCIÓN DE PROCESOS SAS PARA  
EL SOPORTE DE TOMA DE DECISIONES GERENCIALES DEL  
ÁREA DE SOLUCIONES DE INFORMACIÓN DE UNA ENTIDAD  
BANCARIA**

**PRESENTADO POR  
MILUSKA ENMA MARTÍNEZ RODAS  
GESÚ ANDRÉ MEZARINA SILVA**

**ASESOR  
JESÚS NICOLÁS LEÓN LAMAS**

**TESIS  
PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE  
COMPUTACIÓN Y SISTEMAS**

**LIMA – PERÚ**

**2016**



**Reconocimiento  
CC BY**

El autor permite a otros distribuir y transformar (traducir, adaptar o compilar) a partir de esta obra, incluso con fines comerciales, siempre que sea reconocida la autoría de la creación original.

<http://creativecommons.org/licenses/by/4.0/>



**USMP**  
UNIVERSIDAD DE  
SAN MARTÍN DE PORRES

**FACULTAD DE  
INGENIERÍA Y ARQUITECTURA**

**ESCUELA PROFESIONAL DE INGENIERÍA DE COMPUTACIÓN Y  
SISTEMAS**

**DISEÑO DE LA INFRAESTRUCTURA GRID PARA LA  
OPTIMIZACIÓN A LA EJECUCIÓN DE PROCESOS SAS PARA  
EL SOPORTE DE TOMA DE DECISIONES GERENCIALES DEL  
ÁREA DE SOLUCIONES DE INFORMACIÓN DE UNA ENTIDAD**

**BANCARIA**

**TESIS**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE  
COMPUTACIÓN Y SISTEMAS**

**PRESENTADO POR**

**MARTÍNEZ RODAS, MILUSKA ENMA  
MEZARINA SILVA, GESÚ ANDRÉ**

**LIMA – PERÚ**

**2016**



### **Dedicatoria**

El presente trabajo está dedicado a nuestros padres quienes nos apoyaron incondicionalmente en la realización de esta tesis.



### **Agradecimiento**

Expresamos un sincero agradecimiento a la Universidad "San Martín de Porres" por ser una de las mejores universidades en nuestro país y contar con acreditaciones internacionales que permiten estandarizar el nivel de enseñanza a nivel mundial; a nuestros asesores y docentes por enriquecer los conocimientos que hemos adquirido a lo largo de la carrera universitaria.

## ÍNDICE

	Página
<b>RESUMEN</b>	x
<b>ABSTRACT</b>	xi
<b>INTRODUCCIÓN</b>	xii
<b>CAPÍTULO I: MARCO TEÓRICO</b>	15
1.1 Antecedentes	15
1.2 Bases teóricas	19
1.3 Definición de términos básicos	30
<b>CAPÍTULO II: METODOLOGÍA</b>	31
2.1 Materiales	31
2.2 Métodos	31
<b>CAPÍTULO III: DESARROLLO DEL PROYECTO</b>	38
3.1 Análisis de las incidencias para definir la situación actual de la infraestructura	38
3.2 Determinación de la tecnología a utilizar para formular una infraestructura adecuada	45
3.3 Diseño de la infraestructura propuesta SAS Grid	57
3.4 Realización de la migración de procesos SAS actuales del área de Soluciones de Información para la nueva infraestructura SAS Grid	63
<b>CAPÍTULO IV: PRUEBAS Y RESULTADOS</b>	65
4.1 Pruebas	65
4.2 Resultados	65
<b>CAPÍTULO V: DISCUSIÓN Y APLICACIÓN</b>	67
5.1 Discusión	67
5.2 Aplicación	67
<b>CONCLUSIONES</b>	75
<b>RECOMENDACIONES</b>	76
<b>FUENTES DE INFORMACIÓN</b>	77

## Lista de tablas

		Página
Tabla 1	Definición de términos básicos	30
Tabla 2	Tabla de herramientas	31
Tabla 3	Diagrama de entradas, actividades/herramientas, salidas	33
Tabla 4	Diagrama de entradas, actividades/herramientas, salidas análisis de necesidades	35
Tabla 5	Diagrama de entradas, actividades/herramientas, salidas diseño de solución	36
Tabla 6	Diagrama de entradas, actividades/herramientas, salidas	38
Tabla 7	Tabla de pesos en el servidor SAS PSASP04	41
Tabla 8	Procesos del área de Soluciones de Información	41
Tabla 9	Consumo de procesador en la infraestructura actual e los procesos SAS de la entidad bancaria	43
Tabla 10	Ventajas y desventajas de incrementar la potencia del hardware para abastecer la gran demanda de los procesos SAS enviados a ejecutar	44
Tabla 11	Cambio de la infraestructura centralizada por la infraestructura SAS Grid distribuida, la cual permitirá disminuir los tiempos de ejecución de los procesos SAS	44
Tabla 12	Cuadro comparativo entre la opción 1 y opción 2 propuestas	44
Tabla 13	Diagrama de Entradas, Actividades/Herramientas, Salidas de análisis de necesidades	46
Tabla 14	Hardware y software utilizados en infraestructura actual	46
Tabla 15	Proceso PSI_resumen_saldo	47
Tabla 16	Proceso Comportamiento_y_alertas	47
Tabla 17	Proceso Base_espejo_buro	48
Tabla 18	Proceso Estados_financieros	48
Tabla 19	Proceso Applicant_cco	49
Tabla 20	Proceso Applicant_pyme	49
Tabla 21	Proceso Base_parámetros	49

Tabla 22	Proceso Seguimiento_continuo	50
Tabla 23	Proceso Behavior_pyme	50
Tabla 24	Proceso Score_PE_consumo_1_30	51
Tabla 25	Proceso Score_PP_consumo_31-60	51
Tabla 26	Proceso Score_PE_Tarjetas_31-60	51
Tabla 27	Proceso Score_PPPYMERev_31-60	52
Tabla 28	Proceso Score_PPPYMENORev_31-60	52
Tabla 29	Proceso Score_PE_tarjetas_61-90	53
Tabla 30	Proceso Score_PPPYMERev_61-90	53
Tabla 31	Proceso Score_PPPYMENORev_61-90	53
Tabla 32	Proceso Seguimiento_diario	54
Tabla 33	Proceso Tabla Scores_CBM	54
Tabla 34	Proceso Control_de_calidad	55
Tabla 35	Proceso Tabla Control_banca_negocio	55
Tabla 36	Proceso Tabla Tablero_APP_pyme	55
Tabla 37	Diagrama de Entradas, Actividades/Herramientas, Salidas de diseño de solución	58
Tabla 38	Tabla de evaluación de hardware y software a utilizar	58
Tabla 39	Tiempo de ejecución proceso Comportamiento_y_alertas del área de Soluciones de Información	66
Tabla 40	Tiempo de ejecución proceso Base_espejo_buro del área de Soluciones de Información	66
Tabla 41	Tiempo de ejecución proceso PSI_resumen_saldo del área de Soluciones de Información	67
Tabla 42	Tiempo de ejecución proceso Applicant_cco del área de Soluciones de Información	67
Tabla 43	Tiempo de ejecución proceso Estados_financieros del área de Soluciones de Información	68
Tabla 44	Tiempo de ejecución proceso Applicant_pyme del área de Soluciones de Información	68
Tabla 45	Tiempo de ejecución proceso Base_parámetros del área de Soluciones de Información	69
Tabla 46	Tiempo de ejecución proceso Seguimiento_continuo del	70



	área de Soluciones de Información	
Tabla 47	Tiempo de ejecución proceso Behavior_pyme del área de Soluciones de Información	71
Tabla 48	Tiempo de ejecución de los procesos SAS del área de Soluciones de Información	72
Tabla 49	Porcentaje de disminución de la diferencia en minutos entre el tiempo del servidor actual y del servidor SAS Grid	73



## Lista de figuras

		Página
Figura 1	Sistema financiero peruano	16
Figura 2	Pasos para toma de decisiones	18
Figura 3	Ciclo de información de la entidad bancaria	19
Figura 4	Niveles lógicos de una solución grid	21
Figura 5	Diagrama infraestructura SAS Grid	24
Figura 6	Fases de la metodología Kendall & Kendall	25
Figura 7	Fases de la metodología basado en Kendall & Kendall	32
Figura 8	Flujo de actividades	39
Figura 9	Cantidad de caídas por proceso del área de Soluciones de Información del año 2016	40
Figura 10	Cantidad de motivos de caída de los procesos SAS del área de Soluciones de Información	40
Figura 11	Diagrama de la infraestructura actual	56
Figura 12	Diagrama de los procesos SAS de la entidad bancaria	57
Figura 13	Diseño de la infraestructura tecnológica recomendada	62
Figura 14	Distribución de los servidores Grid	62

## Lista de anexos

		Página
Anexo 1	Incidentes SAS 2016	80
Anexo 2	Informe de viabilidad	82
Anexo 3	Consideraciones para la migración del proceso SAS PSI_resumen_saldo del área de Soluciones de Información	87
Anexo 4	Consideraciones para la migración del proceso SAS Comportamiento_y_alertas del área de Soluciones de Información	97
Anexo 5	Consideraciones para la migración del proceso SAS Base_espejo_buro del área de Soluciones de Información	103
Anexo 6	Consideraciones para la migración del proceso SAS Estados_financieros del área de Soluciones de Información	110
Anexo 7	Consideraciones para la migración del proceso SAS Applicant_cco del área de Soluciones de Información	115
Anexo 8	Consideraciones para la migración del proceso SAS Applicant_pyme del área de Soluciones de Información	122
Anexo 9	Consideraciones para la migración del proceso SAS Base_parámetros del área de Soluciones de Información	129
Anexo 10	Consideraciones para la migración del proceso SAS Seguimiento_continuo del área de Soluciones de Información	137
Anexo 11	Consideraciones para la migración del proceso SAS Behavior_pyme del área de Soluciones de Información	147

## RESUMEN

La presente tesis consiste en realizar el diseño de la infraestructura Grid para la optimización de la ejecución de los procesos SAS que apoye a la toma de decisiones gerenciales del área de Soluciones de Información de una entidad bancaria, con lo cual permitirá mejorar la disponibilidad de la información de los reportes analíticos emitidos por los procesos SAS para la alta gerencia en las fechas establecidas.

Se ha realizado el diseño de la solución propuesta en base a la metodología de Kendall & Kendall, el cual sostiene que el ciclo de vida de un sistema consta de 7 fases metodológicas, las cuales tienen como resultado final un entregable por cada fase. Para la presente tesis, se tomaron 4 fases que apoyaron al diseño propuesto que se presentó a la entidad bancaria para mejorar su infraestructura y eficiencia operativa.

Como prueba de la solución se realizó la ejecución de los procesos SAS en un escenario que simuló la infraestructura SAS Grid, en el cual se observó la reducción del tiempo de ejecución en más de un 50 % al tiempo actual.

Finalmente, con los resultados obtenidos se mostró una mejora en los tiempos de ejecución de los procesos SAS que emiten reportes analíticos que ayudará a la entidad bancaria a tomar decisiones gerenciales que apoyen a los objetivos estratégicos y análisis de nuevos productos crediticios.

**Palabras claves:** Infraestructura Grid, optimización, procesos SAS.

## ABSTRACT

The thesis consists in designing the Grid infrastructure for the optimization of the execution of SAS processes that supports management decisions for Information Solution area to a banking entity, which will allow to improve the availability of the Information to the analytical reports issued by SAS processes for the senior management in date.

The proposed solution was based on Kendall & Kendall methodology, which explains that the life cycle of a system consists in 7 methodological phases, which have as final result a deliverable for each phase. For the thesis, 4 phases were taken for the proposed design to the bank in order to improve the infrastructure and operational efficiency.

As a test of the solution, the execution of the SAS processes was performed in a scenario that simulated the SAS Grid infrastructure, in which the execution time was decreased in more than 50% at the current time.

Finally, the results showed an improvement in the execution times of SAS processes to the analytical reports that will help the bank to make management decisions that support the strategic objectives and the analysis of new credit products.

**Key words:** Grid infrastructure, optimization, SAS processes.

## INTRODUCCIÓN

La era tecnológica ha avanzado con el paso del tiempo, lo cual conlleva a actualizar las soluciones tecnológicas con las que se trabaja actualmente. La solución Grid surge con el propósito de minimizar costos haciendo uso de la compartición de recursos de hardware que permiten reducir el tiempo de ejecución de procesos computacionales.

La tecnología SAS permite manejar datos que son extraídos para el análisis de los mismos, el cual funciona en un entorno centralizado lo cual es susceptible a caídas si se llegase a presentar saturación en el servidor.

Por ende, la presente tesis pretende hacer uso de Grid y la tecnología SAS para proponer a la entidad bancaria una infraestructura tecnológica que permita optimizar los tiempos de ejecución de los procesos SAS que son los encargados de emitir reportes analíticos para la toma de decisiones gerenciales, los cuales han presentado problemas por no presentarse en el tiempo establecido.

La solución que se propondrá tiene como objetivo que el área de Soluciones de Información migre a la infraestructura SAS Grid ya que permitirá reducir los tiempos de ejecución haciendo uso del balanceo de carga de trabajo que serán distribuidos por 4 nodos, lo cual convertirá al servidor centralizado y dependiente en un servidor descentralizado e independiente.

Con los reportes emitidos en la fecha establecida por los procesos SAS, la entidad podrá tomar decisiones gerenciales que apoyarán directamente a los objetivos estratégicos que tendrá como finalidad incrementar los ingresos y poder realizar mediante el resultado de los reportes un mejor análisis para crear o aumentar nuevos productos para sus clientes.

## **1. Planteamiento del problema**

Ineficiente infraestructura tecnológica para el soporte en la ejecución de los procesos SAS que emiten reportes analíticos; lo que limita la toma de decisiones gerenciales en el área de soluciones de información de una entidad bancaria.

## **2. Objetivos**

### **2.1. Objetivo general**

Diseñar la infraestructura Grid para la optimización en la ejecución de los procesos SAS para el soporte a la toma de decisiones gerenciales del área de soluciones de información de una entidad bancaria.

### **2.2. Objetivos específicos**

- Analizar por medio de las incidencias, la situación actual de la infraestructura.
- Determinar la tecnología a utilizar para formular una infraestructura adecuada.
- Diseñar la propuesta de infraestructura tecnológica SAS Grid.
- Realizar la migración de los procesos SAS actuales del área de soluciones de información de la entidad bancaria hacia la nueva infraestructura SAS Grid en plataforma AIX.

## **3. Justificación**

Esta solución es muy beneficiosa para la entidad bancaria porque no dependerá de un servidor centralizado, ya que contará con una infraestructura descentralizada bajo la plataforma AIX.

La plataforma descentralizada contará con diferentes nodos los cuales compartirán ciclos de procesamiento, de esta forma si llegara a saturarse uno de los nodos, este no se detendrá y seguirá ejecutando con normalidad los procesos SAS, los cuales finalmente emitirán los reportes analíticos para

el área de soluciones de información, sin generar una caída masiva, pérdidas monetarias y retrasos en la toma de decisiones gerenciales.

#### **4. Limitación**

La limitación que tiene la presente tesis es el tiempo dado que no se pudo llegar a la implementación de la infraestructura SAS Grid en todas las áreas de la entidad bancaria que hacen uso de procesos analíticos.

#### **5. Alcance**

El alcance de la presente tesis abarca los 21 procesos SAS del área de Soluciones de Información, mas no de otras áreas de la entidad bancaria que también utilizan procesos SAS.





# CAPÍTULO I

## MARCO TEÓRICO

### 1.1 Antecedentes

#### 1.1.1 Sistema financiero

El sistema financiero está formado por un conjunto de instituciones y mercados, que tienen como objetivo canalizar el ahorro que generan los prestamistas hacia los prestatarios, así como facilitar y otorgar seguridad al movimiento de dinero y sistema de pagos.

El sistema financiero comprende, tanto los instrumentos o activos financieros, como las instituciones y los mercados financieros, en el cual los intermediarios compran y venden los activos en los mercados financieros (Calvo, Parejo, Rodríguez & Cuervo, 2010).

#### 1.1.2 Sistema financiero peruano

El sistema financiero peruano está conformado por el conjunto de instituciones bancarias, financieras y demás empresas e instituciones de derecho público privado, debidamente autorizadas por la Superintendencia de Banca y Seguro, que operan en la intermediación financiera (Chugnas, 2011).



Figura N° 1: Sistema financiero peruano  
Fuente: Chugnas, 2011

### 1.1.3 Banco de Crédito del Perú

El Banco de Crédito del Perú forma parte del Sistema Financiero Peruano, el cual está regido por la Superintendencia de Banca y Seguros.

Fue llamado durante sus primeros 52 años Banco Italiano, inició sus actividades el 9 de abril de 1889. El 01 de febrero de 1942, se acordó sustituir la antigua denominación social, por la de Banco de Crédito del Perú.

Con el propósito de conseguir un mayor peso internacional, instalaron sucursales en Nassau y en Nueva York, hecho que los convirtió en el único banco peruano presente en dos de las plazas financieras más importantes del mundo.

La expansión de sus actividades creó la necesidad de una nueva sede para la dirección central. Con ese fin, construyeron un edificio de 30,000 m<sup>2</sup>, aproximadamente en el distrito de La Molina.

Luego, con el objetivo de mejorar sus servicios, establecieron la Red Nacional de Tele Proceso, que a fines de 1988 conectaba casi todas las oficinas del país con el computador central de Lima; asimismo, crearon la Cuenta Corriente y Libreta de Ahorro Nacional, e instalaron una extensa red de cajeros automáticos.

Al cumplir 125 años en el mercado local, la institución cuenta con 375 Agencias, más de 1800 cajeros automáticos, más de 5600 Agentes BCP y más de 15000 colaboradores; así como bancos corresponsales en todo el mundo (BCP, 2013).

El Banco del Perú cuenta con 54 áreas que apoyan al buen funcionamiento de la organización. Dentro del área de Administración de Riesgos de Banca Minorista se tiene el área de soluciones de información, el cual será objeto de la investigación.

#### **1.1.4 Toma de decisiones gerenciales**

La toma de decisiones es fundamental en cualquier actividad humana puesto que todos como seres humanos toman decisiones diariamente, para tomar una decisión acertada consta de un proceso de razonamiento constante y focalizado.

Para un gerente que toma decisiones frecuentemente, se le presentan dos tipos de decisiones; las rutinarias que son por lo general decisiones que se toman rápidamente y sin elaborar un proceso detallado de consideraciones y las decisiones que son complejas, críticas o importantes que son las que no pueden ni deben salir mal ya que pueden involucrar ganancias o pérdidas de grandes sumas de dinero para la organización y por ello el gerente deberá hacer un análisis profundo (Amaya, 2010).



**Figura N° 2:** Pasos para toma de decisiones  
**Fuente:** Amaya, 2010

### 1.1.5 Área de Soluciones de Información

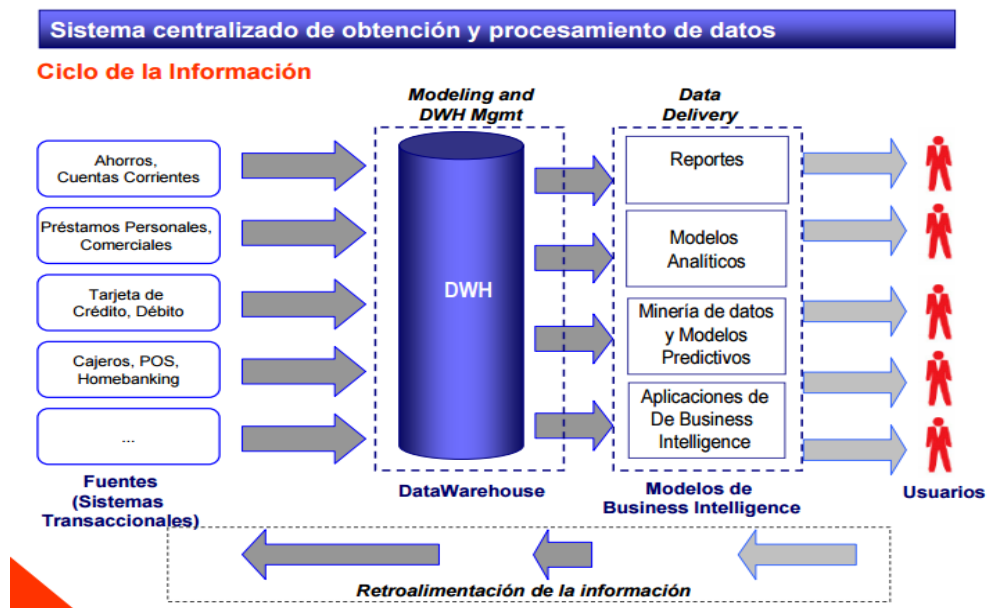
El área de Soluciones de Información es la encargada del modelamiento y metodología de la gestión de riesgo crediticio sobre el comportamiento de los clientes.

Actualmente, los usuarios del área de Soluciones de Información están presentando una gran cantidad de incidencias como caída del servidor de Software de Análisis Estadísticos (SAS), demora en la ejecución de procesos SAS o entrega de reportes fuera de fecha.

Dentro de la amplia infraestructura tecnológica que posee la entidad bancaria en el cual está implicado aproximadamente n servidores, se tiene un servidor SAS bajo la plataforma Windows que es el soporte para ejecutar los procesos SAS enviados por los usuarios del área de Soluciones de Información.

Estos procesos tienen como finalidad extraer data en forma de reportes sumamente importantes para la toma de decisiones del área. Debido a los 22 procesos SAS que son enviados actualmente a este servidor para ser ejecutados, ocasionan que por momentos el servidor se sature y que tome

demasiado tiempo en recuperar la estabilidad, esto se debe a que la infraestructura actual SAS es centralizada por lo que depende de un solo nodo, por ende, no tiene la capacidad de distribuir la carga ni de ejecutar los procesos por prioridad ya que los ejecuta de manera secuencial, es decir por orden de llegada.



**Figura N° 3:** Ciclo de Información de la entidad bancaria

Fuente: Propia

## 1.2 Bases teóricas

Actualmente, las organizaciones que procesan grandes volúmenes de información les toma demasiado tiempo ejecutar sus procesos, es por ello, que sería idóneo que estas organizaciones cuenten con una supercomputadora, ya que de esta manera podrán procesar sus tareas en menor tiempo.

Debido al alto costo de estas supercomputadoras, no es fácil para las organizaciones contar con este equipamiento, es por ello, que se origina el *Grid Computing*, esta solución permite ejecutar las tareas repartiendo la carga de procesamiento en diferentes procesadores de diferentes partes del mundo que pertenezcan a la Grid.

SAS son procesos de análisis estadísticos que se alimenta de la información histórica de datawarehouse para emitir reportes predictivos que son importantes para la toma de decisiones de las organizaciones.

Hoy en día es usado por entidades bancarias, financieras, grandes establecimientos de venta entre otros.

Es indispensable que el servidor SAS tenga alta disponibilidad porque siendo un servidor centralizado es vulnerable a que se sature y presente caídas, por ello nace SAS Grid, el cual es una solución en malla que hace uso de varios nodos para compartir ciclos de procesamiento.

SAS Grid ha ayudado a varias organizaciones a mejorar el rendimiento y tiempos de respuesta.

### **1.2.2 Grid Computing**

Existen varios autores que definen el concepto del Grid computing en la actualidad y cuál es su importancia de los cuales se destacan los siguientes: Grid computing es la compartición de potencia computacional, aprovechando los ciclos de procesamiento no utilizados de cada nodo que pertenecen a la Grid (Enriquez & Ysaías, 2007) .

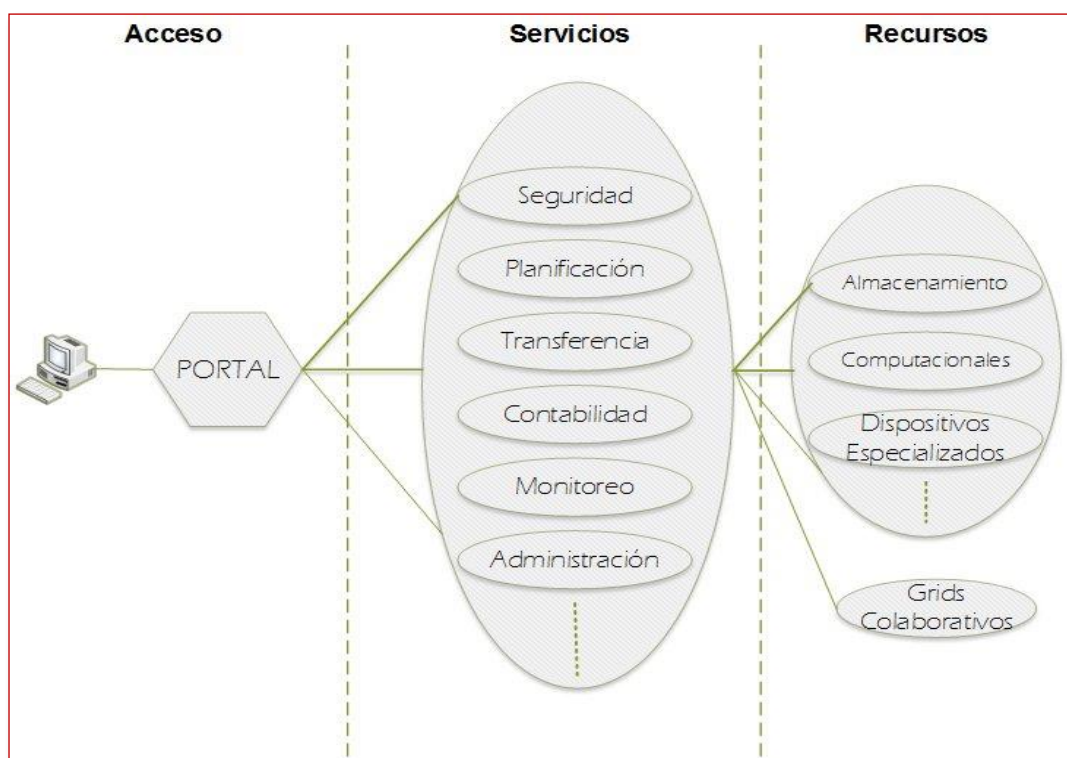
La tecnología de Grid permite usar en forma coordinada diferentes recursos que no están controlados por un ente centralizado en un ambiente distribuido, por lo general heterogéneo en arquitecturas, sistemas operativos, aplicaciones y redes (Isaza & Mendez, 2007).

Por lo tanto, se entiende que *Grid Computing* permite distribuir la carga de procesamiento a otros equipos que se encuentran dentro del grid y que su nivel de procesamiento no se encuentre saturado, esto con la finalidad de minimizar los tiempos de respuesta aprovechando recursos ociosos.

### 1.2.3 Funcionamiento del Grid

Antes de que el usuario pueda hacer uso del grid es necesario que tenga instalado un software en su máquina, lo cual le permitirá el acceso al grid y mediante certificados de seguridad se logrará asegurar la identidad del usuario en la grid, después de la autenticación, el software es proporcionado al usuario para usar el Grid (Gómez, 2014).

A continuación, se muestra los tres niveles lógicos que representa el funcionamiento de un Grid.



**Figura N° 4:** Niveles lógicos de una solución Grid

**Fuente:** Castro, 2006

El primer nivel es el nivel de acceso que resuelve la manera como un usuario final interactúa con el grid, el segundo nivel es el corazón del grid y allí se encuentran los servicios que se puede esperar de un grid y en el último nivel encontramos los recursos propiamente dichos y que deben ser apropiados para ejecutar una cierta tarea.

El nivel de acceso es un portal orientado a aplicaciones. Un grid se construye para resolver problemas y la interacción de un usuario de esta infraestructura debe ser un mecanismo que le permita expresar el problema que desea resolver.

Esta no es una tarea fácil porque mucha de la lógica de distribución va a estar implementada a este nivel. El portal debe ofrecer las herramientas necesarias para que el usuario pueda expresar su conocimiento sobre el problema con el objetivo de maximizar el aprovechamiento del grid para su problema.

El nivel de servicios expresa todo lo que el grid puede hacer por un usuario potencial, es aquí donde las diferentes implementaciones se diferencian y es aquí donde el trabajo es más intenso para lograr la interoperabilidad de las distintas soluciones.

Ante la decisión de seleccionar una solución particular, los administradores de sistemas distribuidos se enfrentan a alternativas que van desde tener el conjunto básico de servicios y a partir de ese conjunto construir la solución más adecuada hasta soluciones empresariales donde todo está ya integrado y donde se define tanto el conjunto de servicios disponibles como la manera particular de interactuar con ellos.

El nivel de los recursos físicos que son los que administran el nivel de servicios y que constituyen en última instancia la plataforma real donde se ejecutarán las aplicaciones (Castro, 2006).

Por lo tanto, existe una dependencia en cada uno de los niveles del grid, ya que para llegar a utilizar los recursos compartidos, computacionales o de almacenamiento, el usuario debe inicialmente ingresar a un portal el cual le ofrecerá las herramientas necesarias para apoyar a la necesidad del cliente.

Luego, el nivel de servicios realizará la tarea de buscar en todos los recursos disponibles dentro de la grid que apoyarán al requerimiento del cliente.



#### **1.2.4 SAS**

El Sistema SAS, llamado Statistical Analysis Software, es un paquete de software que abarca múltiples áreas de trabajo del campo científico, centrándose especialmente en todas las ramas de la Estadística aplicada.

El sistema SAS consta de una serie de módulos, cada uno de ellos orientado a una tarea específica. Entre los más importantes se tienen:

- SAS/BASE, módulo imprescindible sin el cual no funcionan los restantes y que permite el manejo de ficheros y datos, la programación básica y los procedimientos.
- SAS/GRAPH, módulo para la realización de gráficos.
- SAS/STAT, módulo que incluye procedimientos estadísticos complejos.
- SAS/ACCES, SAS/CONNECT y SAS/SHARE, módulos de acceso, conexión y comunicación con bases de datos externas como DB2, ORACLE, SQL SERVER, etc.
- SAS/MINER, módulo de data mining.

SAS consiste en la estructura del lenguaje de programación para manejar los datos, procedimientos sencillos para el análisis de datos, macros para automatizar procesos, programar a medida y un entorno interactivo de ventanas que posibilita editar los programas, analizar, corregir, submitir y ver la salida, tanto numérica como gráfica (Perez, 2013).

#### **1.2.5 SAS Grid**

SAS Grid es un entorno de computación en la que se distribuyen las tareas de computación SAS entre varios ordenadores en una red, todo ello bajo el control de SAS Grid Manager.

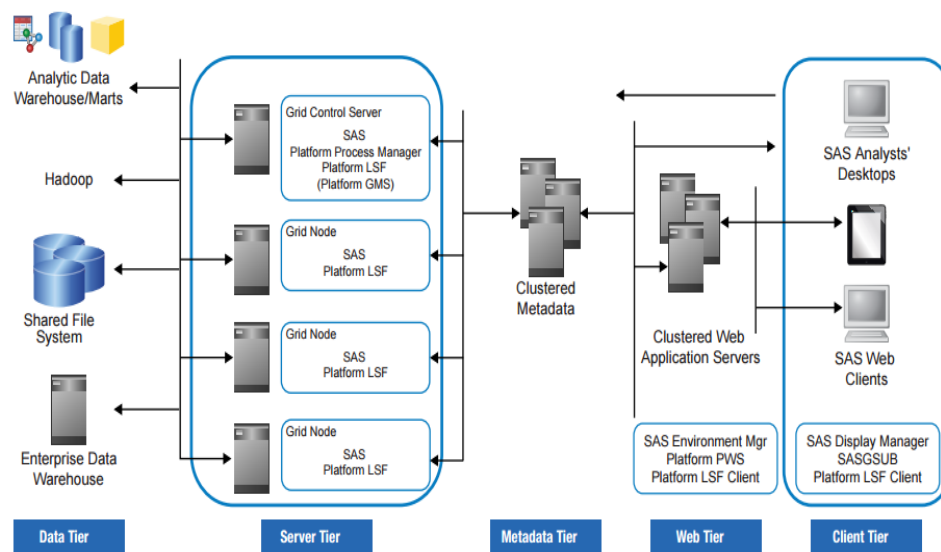
En este entorno, las cargas de trabajo se distribuyen a través de una red de clúster de ordenadores. La distribución de la carga de trabajo permite el balanceo de carga de trabajo con lo cual varios usuarios pueden compartir recursos en un entorno de SAS y el procesamiento acelerado que distribuye subtareas de jobs individuales SAS en conexiones de recursos compartidos.

La Grid permite que las subtareas se ejecuten en paralelo en diferentes partes de la grid lo cual permite la ejecución de los jobs mucho más rápido. También, permite a los usuarios programar los jobs, los cuales son enviados automáticamente a las conexiones de recursos compartidos que se encuentren más liberados.(SAS Institute, 2015).

### 1.2.6 Infraestructura SAS Grid

El siguiente diagrama es una vista conceptual de la arquitectura en red SAS utilizando SAS Grid Manager con la Suite de Plataforma de SAS.

La Suite de Plataforma de SAS es un conjunto de componentes que proporciona eficientemente la asignación de recursos, la gestión de políticas y el equilibrio de carga de trabajo de las solicitudes SAS (SAS Institute, 2015).



**Figura N° 5:** Diagrama Infraestructura SAS Grid  
Fuente: SAS Institute, 2015

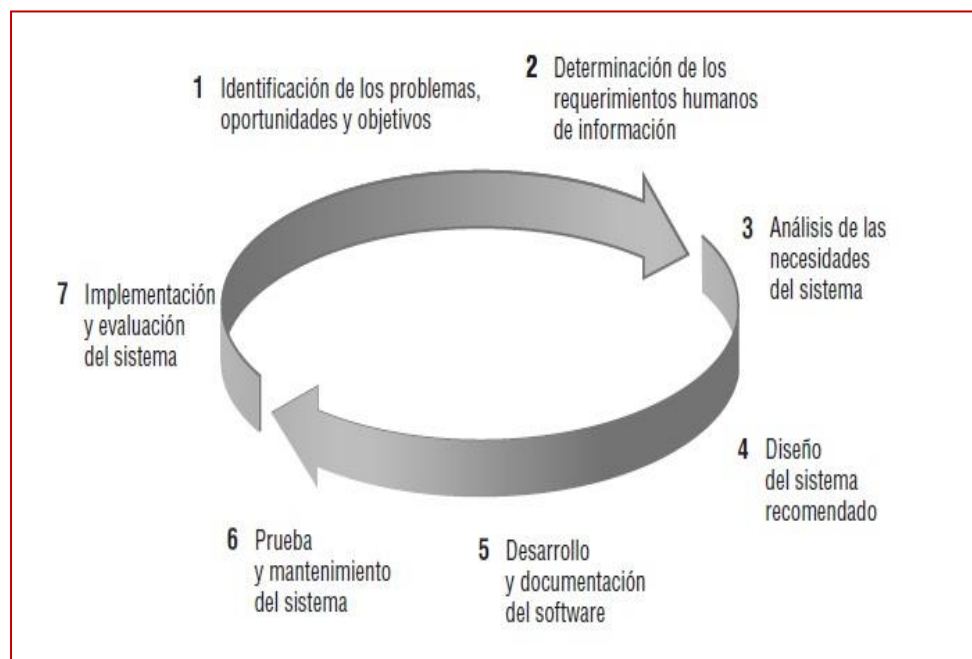
La Suite de Plataforma de SAS incluye los componentes de facilitador de carga compartida, administración del proceso, plataforma de servicios Grid, servicios de Administración Grid y orquestador Grid (SAS Institute, 2015).

### 1.2.7 Metodología de Kendall y Kendall

El ciclo de vida de vida del desarrollo de sistemas es un enfoque por fases para el análisis y el diseño cuya premisa principal consiste en que los sistemas se desarrollan mejor utilizando un ciclo específico de actividades del analista y el usuario (Kendall & Kendall, 2011).

Según la metodología de Kendall & Kendall, el ciclo de vida de un sistema consta de siete partes: siendo la primera la identificación del problema, la segunda identificación de requisitos de información, la tercera es el análisis de las necesidades del sistema, la cuarta es el diseño del sistema recomendado, la quinta desarrollo y documentación del sistema, la sexta prueba y mantenimiento y la última implementación y evaluación.

Cada fase se explica por separado pero nunca se realizan como pasos aislados, más bien es posible que algunas actividades se realicen de manera simultánea, y algunas de ellas podrían repetirse (Delgado & Lozano, 2014).



**Figura N° 6:** Fases de la Metodología Kendall y Kendall  
**Fuente:**Kendall & Kendall, 2011

### **Fase I: Identificación de los problemas, oportunidades y objetivos**

En la primera fase el analista es el encargado de identificar los problemas de la organización, detallarlos, examinar, evaluar las oportunidades y objetivos.

El analista debe identificar y evaluar los problemas existentes en la organización de manera crítica y precisa. Mayormente los problemas son detectados por alguien más y es cuando el analista es solicitado a fin de precisarlos.

Las oportunidades son situaciones que el analista considera susceptibles de mejorar utilizando sistemas de información computarizados, lo cual le da mayor seguridad y eficacia a las organizaciones además de obtener una ventaja competitiva.

El analista debe identificar los objetivos, es decir, el analista debe averiguar lo que la empresa trata de conseguir, se podrá determinar si algunas funciones de las aplicaciones de los sistemas de información pueden contribuir a que el negocio alcance sus objetivos aplicándolas a problemas u oportunidades específicos.

Los usuarios, los analistas y los administradores de sistemas que coordinan el proyecto son los involucrados en la primera fase. Las actividades de esta fase son las entrevistas a los encargados de coordinar a los usuarios, sintetizar el conocimiento obtenido, estimar el alcance del proyecto y documentar los resultados. El resultado de esta fase es un informe de viabilidad que incluye la definición del problema y un resumen de los objetivos. La administración debe decidir si se sigue adelante o si se cancela el proyecto propuesto.

### **Fase II: Determinación de los requerimientos de información**

En esta fase el analista se esfuerza por comprender la información que necesitan los usuarios para llevar a cabo sus actividades. Entre las herramientas que se utilizan para determinar los requerimientos de información de un negocio se encuentran métodos interactivos como las entrevistas, los muestreos, la investigación de datos impresos y la aplicación

de cuestionarios; métodos que no interfieren con el usuario como la observación del comportamiento de los encargados de tomar las decisiones y sus entornos e oficina, al igual que métodos de amplio alcance como la elaboración de prototipos.

Esta fase es útil para que el analista confirme la idea que tiene de la organización y sus objetivos.

Los implicados en esta fase son el analista y los usuarios, por lo general los trabajadores y gerentes del área de operaciones.

El analista necesita conocer los detalles de las funciones del sistema actual como la gente involucrada, la actividad del negocio, el entorno donde se desarrollan las actividades, el momento oportuno y la manera en que se realizan los procedimientos actuales del negocio que se estudia.

Al término de esta fase, el analista debe conocer el funcionamiento del negocio y poseer información muy completa acerca de la gente, los objetivos, los datos y los procedimientos implicados.

### **Fase III: Análisis de las necesidades del sistema**

En esta fase el analista evalúa las dos fases anteriores, usa herramientas y técnicas como el uso de diagramas de flujo de datos para graficar las entradas, los procesos y las salidas de las funciones del negocio en una forma gráfica estructurada.

A partir de los diagramas de flujo de datos se desarrolla un diccionario de datos que enlista todos los datos utilizados en el sistema así como sus respectivas especificaciones.

El analista prepara en esta fase, una propuesta de sistemas que sintetiza sus hallazgos, proporciona un análisis de costo/beneficio de las alternativas y ofrece, en su caso, recomendaciones sobre lo que se debe hacer.

#### **Fase IV: Desarrollo y documentación del software**

En esta fase el analista utiliza la información recopilada en las primeras fases para realizar el diseño lógico del sistema de información.

El analista diseña procedimientos precisos para la captura de datos que aseguran que los datos que ingresen al sistema de información sean correctos.

Facilita la entrada eficiente de datos al sistema de información mediante técnicas adecuadas de diseño de formularios y pantallas. La concepción de la interfaz de usuario forma parte del diseño lógico del sistema de información.

La interfaz conecta al usuario con el sistema y por tanto es sumamente importante. También, incluye el diseño de archivos o bases de datos que almacenarán gran parte de los datos indispensables para los encargados de tomar las decisiones en la organización.

En esta fase el analista interactúa con los usuarios para diseñar la salida en pantalla o impresa que satisfaga las necesidades de información de estos últimos.

Finalmente, el analista debe diseñar controles y procedimientos de respaldo que protejan al sistema y a los datos y producir paquetes de especificaciones de programa para los programadores. Cada paquete debe contener esquemas para la entrada y la salida, especificaciones de archivos y detalles del procesamiento.

#### **Fase VI: Prueba y mantenimiento del sistema**

Antes de poner en funcionamiento el sistema es necesario probarlo puesto que es mucho menos costoso encontrar los problemas antes que el sistema se entregue a los usuarios.

Una parte de las pruebas las realizan los programadores solos, y otra la llevan a cabo de manera conjunta con los analistas de sistemas. Primero se realizan las pruebas con datos de muestra para determinar con precisión cuáles son los problemas y posteriormente se realiza otra con datos reales del sistema actual.

El mantenimiento del sistema de información y su documentación empiezan en esta fase y se llevan de manera rutinaria durante toda su vida útil.

### **Fase VII: Implementación y evaluación del sistema**

Esta es la última fase del desarrollo de sistemas, y aquí el analista participa en la implementación del sistema de información. En esta fase se capacita a los usuarios en el manejo del sistema. Parte de la capacitación la imparten los fabricantes, pero la supervisión de ésta es responsabilidad del analista de sistemas.

Se menciona la evaluación como la fase final del ciclo de vida del desarrollo de sistemas principalmente en áreas del debate. En realidad, la evaluación se lleva a cabo durante cada una de las fases.

El trabajo de sistemas es cíclico, cuando un analista termina una fase del desarrollo de sistemas y pasa a la siguiente, el surgimiento de un problema podría obligar a regresar a la fase previa y modificar el trabajo realizado (Kendall & Kendall, 2011).

### 1.3 Definición de términos básicos

Tabla N° 1: Definiciones básicas

<b>Término</b>	<b>Definición</b>
Administración del proceso:	Es la interfaz utilizada por el marco de programación SAS para controlar la submisión de jobs y administrar cualquier dependencia entre jobs.
Clúster:	Conjunto de ordenadores en una red que se comporta como una única computadora.
Datawarehouse	Es una base de datos que integra información de diferentes fuentes
Diseño:	Es el proceso de planificar, reemplazar o complementar un sistema. Con ello se entiende que para diseñar hay que coordinar y proyectar para realizar un bosquejo que lo que se quiere realizar. (Senn, 1996).
Facilitador de carga compartida:	Este servicio envía todos los jobs, ya sea por el administrador de procesos o directamente por SAS, y devuelve el estado de cada job. También, maneja cualquier recurso y realiza el equilibrio de carga entre máquinas en un entorno de red.
Instrumento o Activo Financiero:	Es el derecho a recibir ingresos futuros por parte del vendedor.
Job:	Es un conjunto de tareas específicas para lograr un resultado.
Orquestador Grid:	Proporciona soporte para realizar un backup de los servicios críticos en un entorno de alta disponibilidad. Ofrece funciones para el control de los servicios críticos, reiniciar servicios si se detienen y comenzar con el servidor backup si fuera necesario.
Plataforma de servicios Grid:	Es un servicio web que proporciona información a los módulos de SAS Grid.
Procesos SAS:	Sentencias escritas en SAS.
Riesgo Crediticio:	Se refiere a la posibilidad de tener grandes pérdidas monetarias por el motivo de que un cliente no cumpla con las obligaciones de crédito a las cuales se comprometió.
SAS:	Software de Análisis Estadísticos
Servicios de Administración Grid:	Proporciona la información de tiempo de ejecución acerca de los jobs para su visualización en SAS.

Fuente: Propia



## CAPÍTULO II METODOLOGÍA

### 2.1 Materiales

Los materiales que se han utilizado para realizar la presente tesis son los siguientes:

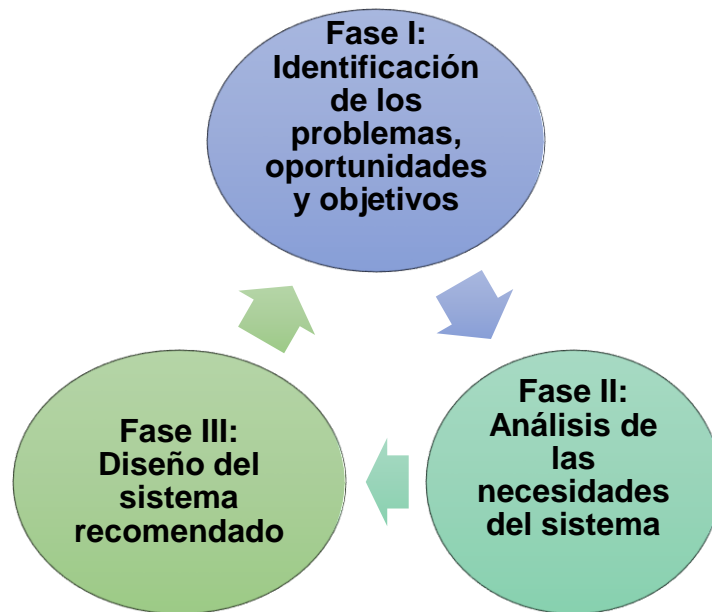
**Tabla N° 2:** Tabla de Herramientas

<b>Herramienta de Documentación</b>		
<b>Software</b>	<b>Versión</b>	<b>Descripción</b>
MS-Word	2013	Herramienta de Microsoft Office que nos permitirá crear documentos de texto requeridos para la gestión y ejecución de la tesis.
MS-Excel	2013	Herramienta de Microsoft Office que nos permitirá revisar los documentos de incidencias entre otros.
MS-Power Point	2013	Herramienta de Microsoft Office que nos permitirá crear las presentaciones (como el alcance, avances, etc) realizadas en toda la ejecución de la tesis.
Zotero	Libre	Herramienta que nos permitirá almacenar fuentes bibliográficas.
Visio	2013	Herramienta que permite graficar los diversos modelos de la tesis.
<b>Herramienta de Comunicación</b>		
<b>Software</b>	<b>Versión</b>	<b>Descripción</b>
GMAIL	Libre	Utilizaremos Gmail para la comunicación con correos electrónicos.
Google Drive	Libre	Herramienta para compartir información en línea y almacenamiento de datos en la nube.

**Fuente:** Propia

### 2.2 Métodos

La metodología Kendall & Kendall cuenta con siete fases, las cuales permiten llegar a la implementación del sistema, debido a que la presente tesis solo se ha definido llegar al diseño de la solución, se escogieron 3 de las 7 fases las cuales se consideran indispensables para el diseño siendo las siguientes fases:



**Figura N° 7:** Fases de la Metodología basado en Kendall & Kendall  
**Fuente:** Propia

La metodología cuenta con tres fases fundamentales que nos permitirá llegar al diseño de la solución, siendo estas fases secuenciales y dependientes una de cada una.

### **2.2.1 Fase I: Identificación de los problemas, oportunidades y objetivos**

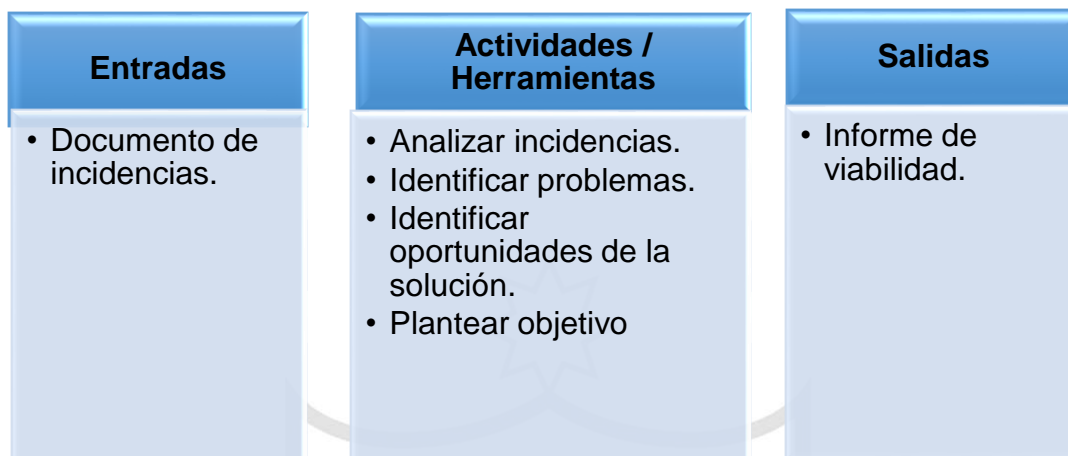
Esta primera fase es considerada indispensable para el correcto desarrollo del proyecto, el analista debe analizar que está ocurriendo para poder identificar cuáles son los problemas que presenta la organización, esto con el apoyo de los documentos de incidencias, una vez el analista tenga identificado los problemas podrá tener una mejor perspectiva del objetivo ya que la realización del proyecto depende de este factor.

El analista debe comprender que es lo que el usuario necesita para culminar satisfactoriamente su trabajo, como interactúa actualmente con el sistema, quienes son los encargados de tomar las decisiones.

El usuario deberá brindar al analista un reporte lo cual servirá para que el analista pueda identificar el problema, objetivo y oportunidades de solución para culminar satisfactoriamente su trabajo.

Al terminar esta fase, el resultado de todo el análisis es un informe de viabilidad el cual contiene la definición del problema, alternativas de solución y objetivo del proyecto a realizar.

**Tabla N° 3:** Diagrama de Entradas, Actividades/Herramientas, Salidas



Fuente: Propia

#### **Documento de Entrada:**

Como documento de entrada se tendrá el documento de incidencias en el cual se encontrará toda la información de los incidentes registrados.

#### **Actividades:**

- **Analizar Incidencias:**

Se realizará el análisis de la información que contiene el archivo de incidencias para finalmente detallar mediante gráficas los problemas encontrados.

- **Identificar Problemas:**

De acuerdo al análisis realizado con el documento de incidencias, se procederá a realizar una lista de los problemas encontrados.

- **Identificar oportunidades de solución:**

En esta parte se detallará las oportunidades de solución que apoyen a la solución del problema.

- **Plantear objetivo:**

Seguidamente, con la información recaudada se deberá plantear cuál es el objetivo como proyecto que apoye a la solución de la problemática.

**Documento de Salida:**

Como documento de salida, se tendrá el informe de viabilidad el cual contendrá la definición del problema, alternativas de solución y objetivo del proyecto a realizar.

**2.2.2 Fase II: Análisis de las necesidades del sistema**

En esta fase, el analista tendrá como referencia el documento de viabilidad culminada en la fase I, el cual permitirá tener una perspectiva de lo que se quiere llegar a realizar, al final de la fase se tendrá la capacidad de uso y la utilidad de los sistemas actuales, es necesario tener información del hardware y software que interactúan con el usuario para realizar el correcto análisis de la situación actual.

El objetivo es identificar y detallar cómo interactúan los flujos de datos con el sistema actual, los cuales permiten al analista ilustrar a los sistemas de una manera gráfica y estructurada para poder entender cómo trabaja el sistema.

Finalmente, el analista elaborará el diagrama de flujo de datos y el diagrama de infraestructura de hardware de como se viene trabajando en la actualidad.

**Tabla N° 4:** Diagrama de Entradas, Actividades/Herramientas, Salidas de análisis de necesidades

Entradas	Actividades / Herramientas	Salidas
<ul style="list-style-type: none"> <li>• Informe de viabilidad.</li> <li>• Documento de flujo de datos</li> </ul>	<ul style="list-style-type: none"> <li>• Identificar el hardware y software existente.</li> <li>• Analizar el flujo de datos.</li> <li>• Elaborar la infraestructura de hardware actual.</li> </ul>	<ul style="list-style-type: none"> <li>• Diagrama de la infraestructura de hardware actual.</li> <li>• Diagrama de flujo de datos.</li> </ul>

Fuente: Propia

#### **Documento de Entrada:**

Como documento de entrada se tendrá el informe de viabilidad culminado en la fase I y el documento de flujo de datos que será brindado por el área de sistemas.

#### **Actividades:**

- **Identificar el hardware y software existente:**

Se deberá consultar al especialista del área encargada cuál es el hardware y software con el que trabaja la infraestructura.

- **Analizar el flujo de datos:**

Se deberá extraer del archivo flujo de datos, el flujo de información con la cual se trabajará.

- **Elaborar la infraestructura de hardware actual**

Con la información anterior, se realizará el diseño de la infraestructura del hardware con la que se viene trabajando actualmente.

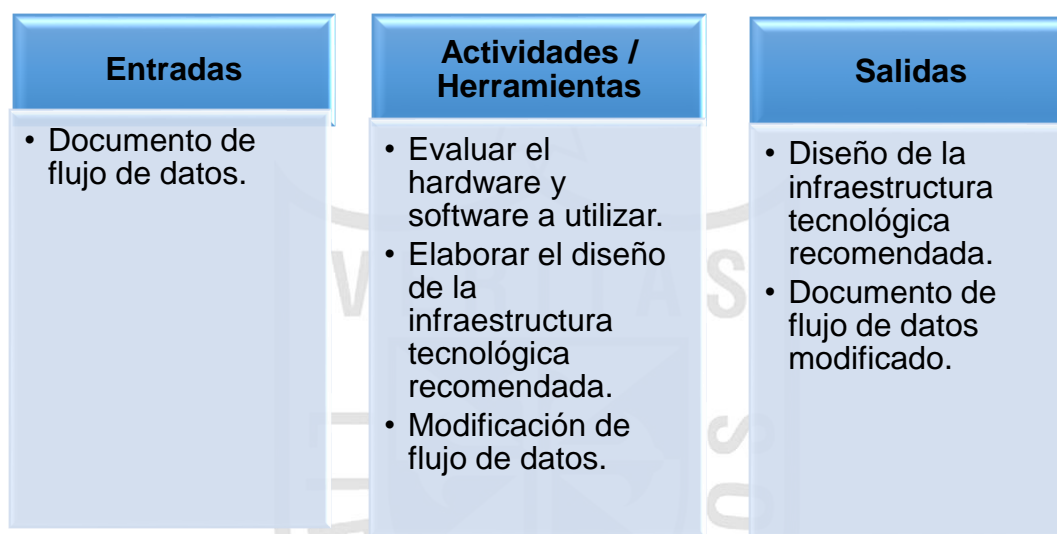
#### **Documento de Salida:**

Como documento de salida, se tendrá el diagrama de infraestructura actual y el diagrama de flujo de datos.

### 2.2.3 Fase III: Diseño del sistema recomendado

Con la Información recaudada en la fase II, el analista deberá evaluar el hardware y software a utilizar para realizar satisfactoriamente el diseño de la solución recomendada, así como la modificación del documento de flujo de datos para finamente obtener el flujo de datos modificado junto con el digrama de la solución recomendada (Kendall & Kendall, 2011).

**Tabla N° 5:** Diagrama de Entradas, Actividades/Herramientas, Salidas de diseño de solución



Fuente: Propia

#### Documento de Entrada:

Como documento de entrada se tendrá el documento flujo de datos culminado en la fase II.

#### Actividades:

- **Evaluar el hardware y software a utilizar:**

Se deberá evaluar el hardware y software de acuerdo a la tecnología recomendada.

- **Modificación de flujo de datos**

Con la información del documento de flujo de datos, se procederá a realizar la adecuación del código para que sea compatible con la tecnología recomendada.

**Documento de Salida:**

Como documento de salida, se tendrá el diseño de la infraestructura tecnológica recomendada y el documento de flujo de datos modificado.



## CAPÍTULO III DESARROLLO DEL PROYECTO

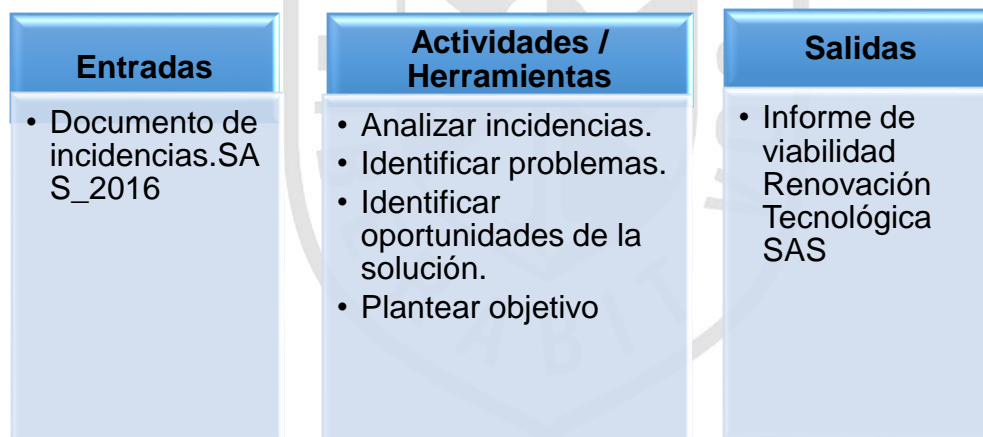
Para el desarrollo del proyecto se tendrá como referencia el documento de incidencias INCIDENTES\_SAS\_2016, aprobado por el área de sistemas de la entidad bancaria, el cual contiene información de todos los procesos SAS que fueron enviados al servidor pero no fueron completados satisfactoriamente.

### 3.1 Análisis de las incidencias para definir la situación actual de la infraestructura

#### Fase I: Identificación de los problemas, oportunidades y objetivos

De acuerdo, a nuestra metodología, para la fase I se seguirán los siguientes lineamientos:

**Tabla N° 6:** Diagrama de Entradas, Actividades/Herramientas, Salidas



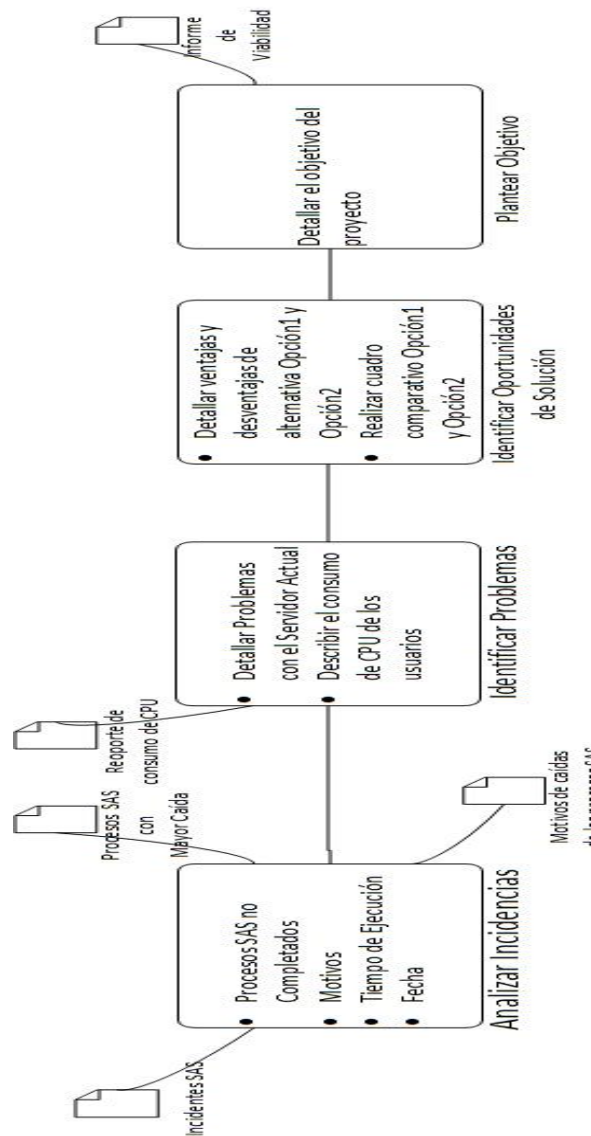
Fuente: Propia

#### Documento de Entrada:

El documento de Entrada INCIDENTES\_SAS\_2016.xlsx (Ver anexo 1), contiene toda la información de los incidentes registrados por los usuarios y que tienen un tiempo histórico de cuatro meses correspondientes al año 2016.



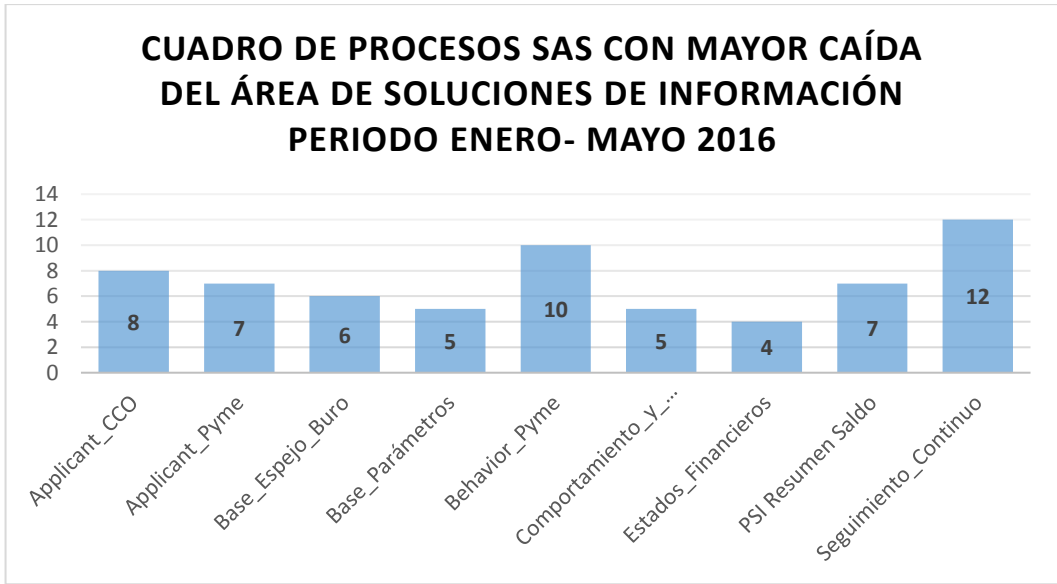
**Actividades:**



**Figura N° 8:** Flujo de actividades  
**Fuente:** Propia

**• Analizar Incidencias:**

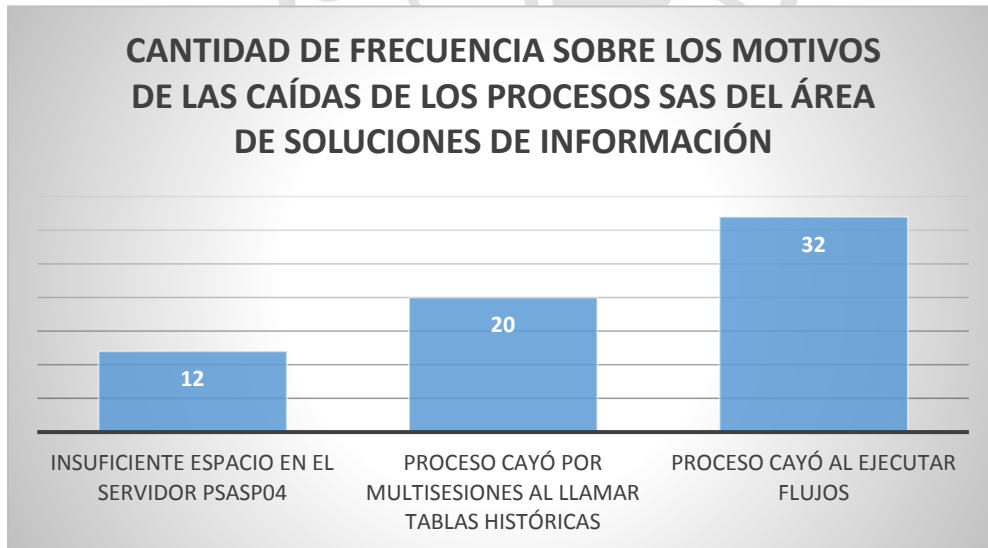
En el archivo Incidencias\_SAS\_2016 se observó los procesos SAS que no fueron ejecutados en su totalidad, se tiene información del motivo por el cual no se culminó satisfactoriamente, el tiempo y la fecha en que el proceso estuvo en ejecución en el servidor SAS antes de caer. Se ha realizado el análisis con los procesos que corresponde al área de soluciones de información durante los meses de enero a abril del presente año, que se detalla a continuación:



**Figura N° 9:** Cantidad de caídas por proceso del área de Soluciones de Información del año 2016

**Fuente:** Propia

Conclusión: Mediante el gráfico, se observa que el proceso con mayor caída en el área de soluciones de información es el proceso de Seguimiento\_Continuo, el cual ha presentado 12 caídas durante los meses de enero a mayo, esto ocasiona demora en la entrega del reporte que emite el proceso para la gerencia.



**Figura N° 10:** Cantidad de motivos de caída de los procesos SAS del área de Soluciones de Información

**Fuente:** Propia

Conclusión: Mediante el gráfico, se observa que la mayor concurrencia en los motivos de las caídas de los procesos del área de soluciones de información se debe a que el proceso cae al ejecutar los flujos de los programas SAS.

Otra información destacada que se debe tomar en consideración, es la cantidad de procesos SAS por área y el peso que ocupan al realizar las ejecuciones en el servidor sas PSASP04.

**Tabla N° 7:** Tabla de Pesos en el servidor SAS PSASP04

Área	Procesos SAS	Peso en Disco
Soluciones de Información	9	8729 GB
Cobranzas	10	307 GB
Riesgo Banca Mayorista	7	1479 GB
Riesgo Banca Minorista	3	6019 GB

Fuente: Propia

También, se tomará en el análisis la información del tiempo de ejecución de los procesos SAS del área de soluciones de información:

**Tabla N° 8:** Procesos del Área de Soluciones de Información

Procesos SAS	Área	Tiempo de ejecución actual
Applicant_cco	SINF	05:27:10 hh:mm:ss
Applicant_pyme	SINF	06:23:32 hh:mm:ss
Base_espejo_buro	SINF	07:15:14 hh:mm:ss
Base_parametros	SINF	06:52:50 hh:mm:ss
Behavior_pyme	SINF	12:42:08 hh:mm:ss
Comportamiento_y_alertas	SINF	08:37:24 hh:mm:ss
Estados_financieros	SINF	04:10:35 hh:mm:ss
PSI_resumen_saldo	SINF	12:25:28 hh:mm:ss
Seguimiento_continuo	SINF	09:40:45 hh:mm:ss

Fuente: Propia

- **Identificar Problemas:**

De acuerdo al análisis de las incidencias, se tiene que el motivo más frecuente por el cual los procesos SAS no se ejecutan satisfactoriamente es “Proceso cayó al ejecutar flujos”, esto quiere decir que por la baja disponibilidad del servidor SAS no se pudo culminar con la ejecución de los proceso SAS.

El segundo motivo más frecuente consiste en la caída de procesos por multisesiones al llamar tablas históricas”, debido a que las sesiones de los procesos SAS son limitados en el servidor centralizado y no maneja un balanceo de carga para los procesos.

El último motivo “Insuficiente espacio en el servidor psasp04”, se debe a las limitaciones del hardware del servidor que ejecutan los procesos SAS.

Los motivos mencionados ocasionan una demora en la ejecución de los procesos SAS, por ende retrasa la emisión de reportes analíticos fundamentales para la toma de decisiones estratégicas del área de soluciones de información.

Finalmente, se detallan los problemas identificados con el análisis de los motivos de caída del Servidor SAS:

- Servidor centralizado.
- No existe un servidor de Respaldo.
- Saturación del servidor SAS.
- No existe balanceo de Carga.
- Demora en la ejecución de los procesos SAS.
- No existe algoritmo de priorización de los procesos SAS.

Analizando los pasos anteriores, se concluye que los problemas que se están presentando se deben a una ineficiente infraestructura tecnológica para el soporte a la ejecución de los procesos SAS.

En la siguiente imagen, se muestra los procesos SAS que se encuentran ejecutando de todas las áreas de la entida bancaria:

**Tabla N° 9:** Consumo de procesador en la infraestructura actual de los procesos SAS de la entidad bancaria

	UserID	USUARIO	AREA	CPU_Horas	PorcCPU
1	BCPDOM\S35503	Fiorella Tenorio	SINF	11.82	31.94
2	BCPDOM\S48904	Boris Sinche	SINF	3.83	10.34
3	BCPDOM\S42396	Paolo Ugarte M.	Cobranzas	3.81	9.92
4	BCPDOM\S11250	Jimmy Farfan	SINF	3.80	9.43
5	BCPDOM\S09364	Mabel Gomez	Cobranzas	2.16	7.97
6	BCPDOM\S22934	Ivan Mendoza	SINF	2.02	7.88
7	BCPDOM\S48477	Ricardo Mini	SINF	1.54	6.37
8	BCPDOM\S42396	Paolo Ugarte M.	Cobranzas	1.11	5.56
9	BCPDOM\S38748	Pablo Garavito	RBM	0.92	3.87
10	BCPDOM\S09364	Mabel Gomez	Cobranzas	0.53	2.21

Fuente: Propia

Conclusión: Se observa que los procesos SAS consumen el 95.49 % del procesador. Actualmente, como solución, los usuarios cancelan los procesos que no son prioritarios para dar preferencia a los que son indispensables.

- **Identificar oportunidades de solución:**

Con los problemas identificados, se plantearon dos alternativas de solución, las cuales se detallarán a continuación:

Opción 1:

Incrementar la potencia del hardware para que pueda abastecer la gran demanda de los procesos SAS enviados a ejecutar.

Opción 2:

Cambiar la infraestructura centralizada por la infraestructura SAS Grid distribuida, la cual permitirá disminuir los tiempos de ejecución de los procesos SAS.

**Tabla 10:** Ventajas y Desventajas de incrementar la potencia del hardware para que pueda abastecer la gran demanda de los procesos SAS enviados a ejecutar

<b>Opción 1: Incrementar la potencia del hardware para que pueda abastecer la gran demanda de los procesos SAS enviados a ejecutar</b>	
<b>Ventajas:</b>	<b>Desventajas:</b>
<ul style="list-style-type: none"> <li>• Se ejecutarán mayor cantidad de procesos SAS.</li> </ul>	<ul style="list-style-type: none"> <li>• No hay algoritmo de priorización de ejecución de los procesos.</li> <li>• No existe servidor de respaldo.</li> <li>• Servidor centralizado</li> </ul>

**Fuente:** Propia

**Tabla 11:** Cambiar la infraestructura centralizada por la infraestructura SAS Grid distribuida, la cual permitirá disminuir los tiempos de ejecución de los procesos SAS

<b>Opción 2: Cambiar la infraestructura centralizada por la infraestructura SAS Grid distribuida, la cual permitirá disminuir los tiempos de ejecución de los procesos SAS</b>	
<b>Ventajas:</b>	<b>Desventajas:</b>
<ul style="list-style-type: none"> <li>• Alta Disponibilidad de la información.</li> <li>• Balanceo de carga de trabajo.</li> <li>• Disminución de los tiempos de ejecución de los procesos SAS.</li> <li>• Servidor de respaldo</li> <li>• Es flexible y escalable.</li> <li>• Algoritmo de priorización de procesos SAS</li> </ul>	<ul style="list-style-type: none"> <li>• Incrementa el uso de Jobs.</li> </ul>

**Fuente:** Propia

El siguiente cuadro muestra la comparación entre la opción 1 y opción 2:

**Tabla N° 12:** Cuadro Comparativo entre la Opción 1 y Opción 2 propuestas

<b>OPCIÓN 1</b>	<b>OPCIÓN 2</b>
<ul style="list-style-type: none"> <li>• Poca Disponibilidad: Si bien se incrementa la potencia de hardware seguirá siendo un servidor centralizado latente a caídas.</li> </ul>	<ul style="list-style-type: none"> <li>• Alta Disponibilidad: La solución SAS GRID por ser un sistema distribuido ofrece alta disponibilidad.</li> </ul>
<ul style="list-style-type: none"> <li>• Carga de procesamiento Centralizada: El servidor actual esta bajo la plataforma Windows lo cual no permite el balanceo de carga.</li> </ul>	<ul style="list-style-type: none"> <li>• Balanceo de carga: El Servidor Grid Manager será el encargado de distribuir la carga entre la cantidad de nodos que tenga en su infraestructura y bajo la plataforma AIX.</li> </ul>

<ul style="list-style-type: none"> <li>• Servidor Centralizado: Es dependiente de un único Servidor el cual ejecuta los procesos SAS y si llegase a caer se paralizarían la ejecución de los procesos SAS.</li> </ul>	<ul style="list-style-type: none"> <li>• Servidor Distribuido: A diferencia de SAS que es un servidor centralizado, SAS Grid no depende de un solo servidor, sino tiene servidor de respaldo, y servidores encargados a la ejecución de los procesos SAS que permiten la compartición de recursos computacionales.</li> </ul>
	<ul style="list-style-type: none"> <li>• Mejora en los tiempos de ejecución de procesos SAS: Con el balanceo de carga y la alta disponibilidad se logrará minimizar los tiempos de ejecución de los procesos SAS.</li> </ul>
	<ul style="list-style-type: none"> <li>• Flexibilidad y Escalabilidad: Pueden añadirse servidores en la infraestructura, a medida que la carga de trabajo se incremente, sin afectar al resto de los servidores.</li> </ul>
	<ul style="list-style-type: none"> <li>• Algoritmo de priorización de ejecución de procesos SAS.</li> </ul>

Fuente: Propia

Por lo tanto, con los factores expuestos es factible la opción 2 porque ofrece mejores beneficios que apoyan a los objetivos estratégicos del área de soluciones de información.

- **Plantear objetivo:**

Diseñar la infraestructura sas grid que optimice el tiempo de ejecución de los procesos SAS.

### 3.2 Determinación de la tecnología a utilizar para formular una infraestructura adecuada

#### Fase II: Análisis de las necesidades del sistema

De acuerdo, a la metodología a seguir Kendall & Kendall, para la fase II se seguirán lo siguientes lineamientos:

**Tabla N° 13:** Diagrama de Entradas, Actividades/Herramientas, Salidas de análisis de necesidades

Entradas	Actividades / Herramientas	Salidas
<ul style="list-style-type: none"> <li>Informe de viabilidad.</li> <li>Documento de procesos SAS</li> </ul>	<ul style="list-style-type: none"> <li>Identificar el hardware y software existente.</li> <li>Detallar los procesos SAS.</li> <li>Elaborar la infraestructura tecnológica actual</li> </ul>	<ul style="list-style-type: none"> <li>Diagrama de la infraestructura actual.</li> <li>Diagrama de procesos SAS.</li> </ul>

Fuente: Propia

**Documento de Entrada:**

Como documento de entrada tenemos el informe de viabilidad realizado en la fase I y el documento de procesos SAS, el cual contiene información de todos los procesos SAS que son enviados al servidor actual.

**Actividades:**

• **Identificar el hardware y software existente**

Con la información proporcionada por el área de sistemas de la entidad bancaria, se identifica el hardware y software existentes utilizados en la infraestructura actual para la ejecución de los procesos SAS, que se detalla en la siguiente tabla:

**Tabla N° 14:** Tabla de hardware y software utilizados en Infraestructura Actual

<b>HARDWARE</b>	CPU core 2 Quad con 8 GB RAM, 1 TB de Disco Duro
<b>SOFTWARE</b>	Sistema Operativo: Windows Server 2008 R2 Base de datos: Oracle

Fuente: Propia



- **Detallar procesos SAS de la entidad bancaria ejecutados en el servidor**

Seguidamente, se describe cada uno de los procesos SAS de la entidad bancaria que son los siguientes:

**Proceso 1: Proceso PSI\_resumen\_saldo:**

Descripción: Analizar la estabilidad de las variables generadas en el proceso de Resumen de saldos.

Se trata de un control interno del área para validación de los procesos.

Generan salidas separadas por el riesgo de perder información.

**Tabla N° 15:** Tabla Proceso PSI\_resumen\_saldo

Responsable de Ejecución:	Jimmy Farfán
Usuario Final de datos generados:	Soluciones de Información
Area:	Soluciones de información
Nivel de Criticidad:	Media
Frecuencia de Ejecución:	Mensual (Primera semana de cada mes).

Fuente: Propia

**Proceso 2: Proceso Comportamiento\_y\_alertas**

Descripción: Indicadores de Métricas sobre comportamientos de los clientes.

Genera dos tablas una para comportamientos y otra de alertas para los clientes mayoristas.

**Tabla N° 16:** Tabla Proceso Comportamiento\_y\_alertas

Responsable de Ejecución:	Iván Mendoza
Usuario Final de datos generados:	Metodologías
Área:	Soluciones de Información
Nivel de Criticidad:	Media
Frecuencia de Ejecución:	Mensual (Primera semana de cada mes).

Fuente: Propia

**Proceso 3: Proceso Base\_espejo\_buro**

Descripción: Generar un score a todo el sistema financiero. Son dos archivos sas secuenciados, cuyo resultado de cada uno, se valida.

La validación se realiza paso a paso. Se ejecuta uno, se valida, y si está ok, continúa el próximo.

**Tabla N° 17:** Tabla Base\_espejo\_buro

Responsable de Ejecución:	Fiorella Tenorio
Usuario Final de datos generados:	Metodologías
Área:	Soluciones de Información
Nivel de Criticidad:	Media
Frecuencia de Ejecución:	Mensual

Fuente: Propia

#### **Proceso 4: Proceso Estados\_financieros**

Descripción: Se calcula mes a mes con ventana de 12 a 24 meses. El universo es proporcionado con RBM.

Se calcula mes a mes con ventana de 12 a 24 meses. El universo es proporcionado con RBM.

Dependencia de las alertas (Exp BCP, Resumen Saldo, Leasing, etc.)

**Tabla N° 18:** Tabla Procesos Estados\_financieros

Responsable de Ejecución:	Iván Mendoza
Usuario Final de datos generados:	RBM, MMGR, ADR(Equipo de Dante Tosso)
Área:	Soluciones de Información
Nivel de Criticidad:	Alta
Frecuencia de Ejecución:	Mensual

Fuente: Propia

#### **Proceso 5: Proceso Applicant\_cco**

Descripción: Se calcula mes a mes el Score (teórico) que indica si a un cliente se le otorga o no un préstamo. La salida es un score a nivel de solicitud.

Son tres scripts .sas que se ejecutan encadenados. El primero copia las tablas de oracle, el segundo calcula los scores y el último guarda los resultados en oracle.

**Tabla N° 19:** Proceso Applicant\_cco

Responsable de Ejecución:	Fiorella Tenorio
Usuario Final de datos generados:	Metodologías
Área:	Soluciones de Información
Nivel de Criticidad:	Alta
Frecuencia de Ejecución:	Mensual

Fuente: Propia

**Proceso 6: Proceso Applicant\_pyme**

Descripción: Proceso de evaluación crediticia para Pyme (Tarjetas de Crédito Pyme, Capital de trabajo y Activos Fijos). Se trata de dos procesos por separado. Por un lado se generan tablas en Oracle, luego se tomarán para realizar los primeros scores. El resultado del primer proceso se almacena en Oracle. Luego un segundo proceso, realiza actualizaciones en Oracle y vuelve a realizar nuevos cálculos. Finalmente se guardan todos los resultados en Oracle.

**Tabla N° 20:** Proceso Applicant\_pyme

Responsable de Ejecución:	Boris Sinche
Usuario Final de datos generados:	Metodologías
Nivel de Criticidad:	Alta
Frecuencia de Ejecución:	Mensual

Fuente: Propia

**Proceso 7: Proceso Base\_parámetros**

Descripción: Proceso para asignación de PD, LAD y LTD para los clientes banca minorista. A nivel de cuenta. Se trata de seis scripts diferentes, uno por producto, independientes entre sí.

**Tabla N° 21:** Proceso Base\_parámetros

Responsable de Ejecución:	Ricardo Miní
Usuario Final de datos generados:	Gestión Global del Riesgo + RBM
Área:	Soluciones de Información
Nivel de Criticidad:	Alta
Frecuencia de Ejecución:	Mensual

Fuente: Propia

### **Proceso 8: Proceso Seguimiento\_Continuo**

Descripción: Genera Reporte para gerencias, productos y riesgos para seguimiento de los modelos estadísticos. Se trata de seis procesos independientes. Uno por cada producto.

**Tabla N° 22:** Tabla Seguimiento\_Continuo

Responsable de Ejecución:	Fiorella Tenorio
Usuario Final de datos generados:	Gerencia de Riesgo
Área:	Soluciones de Información
Nivel de Criticidad:	Alta
Frecuencia de Ejecución:	Mensual

Fuente: Propia

### **Proceso 9: Proceso Behavior\_pyme**

Descripción: Calificar el comportamiento de los clientes Pyme. Para cuando ya el cliente tiene un producto adquirido. Analiza el mes a mes el comportamiento del cliente sobre ese producto.

Cinco valores con PD a nivel cliente, productos, calificadores de trabajo, activo fijo y revolventes tarjetas de pyme, un consolidado final a nivel de producto.

**Tabla N° 23:** Tabla Behavior\_pyme

Responsable de Ejecución:	Boris Sinche
Usuario Final de datos generados:	RBM
Área:	Riesgo Banca Minorista
Nivel de Criticidad:	Alta
Frecuencia de Ejecución:	Mensual

Fuente: Propia

### **Proceso 10: Proceso Score\_PE\_consumo\_1\_30**

Descripción: Monitoreo de calibrado de modelos, scores de cobranzas, seguimiento de variables y modelo. La salida son tablas SAS y se exporta a Excel. También hay salidas en PDF.

**Tabla N° 24:** Tabla Proceso Score\_PE\_consumo\_1\_30

Responsable de Ejecución:	Mabel Gómez
Usuario Final de datos generados:	Cobranzas y Comité de Modelos
Área:	Cobranzas
Nivel de Criticidad:	Media
Frecuencia de ejecución :	Mensual, el 15 de cada mes

Fuente: Propia

### **Proceso 11: Score\_PP\_consumo (Tramo 31-60)**

Descripción: Monitoreo de calibrado de modelos, scores de cobranzas, seguimiento de variables y modelo. La salida son tablas SAS y se exporta a Excel. También hay salidas en PDF.

**Tabla N° 25:** Tabla Score\_PP\_consumo\_31-60

Responsable de Ejecución:	Mabel Gómez
Usuario Final de datos generados:	Cobranzas y Comité de Modelos
Área:	Cobranzas
Nivel de Criticidad:	Media
Frecuencia de Ejecución:	Mensual, el 15 de cada mes.

Fuente: Propia

### **Proceso 12: Score\_PE\_tarjetas (Tramo 31-60)**

Descripción: Monitoreo de calibrado de modelos, scores de cobranzas, seguimiento de variables y modelo. La salida son tablas SAS y se exporta a Excel.

También hay salidas en PDF. Este proceso se ejecuta posterior al proceso PP\_Consumo, porque utiliza tablas generadas.

**Tabla N° 26:** Tabla Score\_PE\_tarjetas\_31-60

Responsable de Ejecución:	Mabel Gómez
Usuario Final de datos generados:	Cobranzas y Comité de Modelos
Área:	Cobranzas
Nivel de Criticidad:	Media
Frecuencia de Ejecución:	Mensual, el 15 de cada mes.

Fuente: Propia

### **Proceso 13: Score PPPYMERev (Tramo 31-60)**

Descripción: Monitoreo de calibrado de modelos, scores de cobranzas, seguimiento de variables y modelo. La salida son tablas SAS y se exporta a Excel. También hay salidas en PDF. Este proceso se ejecuta posterior al proceso Score\_PPConsumo, porque utiliza tablas generadas.

**Tabla N° 27:** Tabla Score\_PPPYMERev\_31-60

Responsable de Ejecución:	Mabel Gómez
Usuario Final de datos generados:	Cobranzas y Comité de Modelos
Área:	Cobranzas
Nivel de Criticidad:	Media
Frecuencia de Ejecución:	Mensual, el 15 de cada mes.

Fuente: Propia

### **Proceso 14: Score PPPYMENORev (Tramo 31-60)**

Descripción: Monitoreo de calibrado de modelos, scores de cobranzas, seguimiento de variables y modelo. La salida son tablas SAS y se exporta a Excel. También hay salidas en PDF.

Este proceso se ejecuta posterior al proceso Score\_PPConsumo, porque utiliza tablas generadas.

**Tabla N° 28:** Tabla Score\_PPPYMENORev\_31-60

Responsable de Ejecución:	Mabel Gómez
Usuario Final de datos generados:	Cobranzas y Comité de Modelos
Área:	Cobranzas
Nivel de Criticidad:	Media
Frecuencia de Ejecución:	Mensual, el 15 de cada mes.

Fuente: Propia

### **Proceso 15: Score PE\_Tarjetas (61-90)**

Descripción: Monitoreo de calibrado de modelos, scores de cobranzas, seguimiento de variables y modelo. La salida son tablas SAS y se exporta a Excel. También hay salidas en PDF.

Este proceso se ejecuta posterior al proceso PP\_Consumo Tramo 61-90, porque utiliza tablas generadas.

**Tabla N° 29:** Tabla Score\_PE\_tarjetas (Tramo 61-90)

Responsable de Ejecución:	Mabel Gómez
Usuario Final de datos generados:	Cobranzas y Comité de Modelos
Área:	Cobranzas
Criticidad:	Media
Frecuencia de Ejecución:	Mensual, el 15 de cada mes.

**Fuente:** Propia

### **Proceso 16: Score\_PPPYMERev (Tramo 61-90)**

Descripción: Monitoreo de calibrado de modelos, scores de cobranzas, seguimiento de variables y modelo. La salida son tablas SAS y se exporta a Excel. También hay salidas en PDF.

Este proceso se ejecuta posterior al proceso Score\_PPConsumo Tramo 61 - 90, porque utiliza tablas generadas.

**Tabla N° 30:** Tabla Score\_PPPYMERev\_61-90

Responsable de Ejecución:	Mabel Gómez
Usuario Final de datos generados:	Cobranzas y Comité de Modelos
Área:	Cobranzas
Nivel de Criticidad:	Media
Frecuencia de Ejecución:	Mensual, el 15 de cada mes.

**Fuente:** Propia

### **Proceso 17: Score\_PPPYMENORev (Tramo 61-90)**

Descripción: Monitoreo de calibrado de modelos, scores de cobranzas, seguimiento de variables y modelo. La salida son tablas SAS y se exporta a Excel. También hay salidas en PDF.

Este proceso se ejecuta posterior al proceso Score\_PPConsumo, porque utiliza tablas generadas.

**Tabla N° 31:** Tabla Score\_PPPYMENORev\_61-90

Responsable de Ejecución:	Mabel Gómez
Usuario Final de datos generados:	Cobranzas y Comité de Modelos
Área:	Cobranzas
Nivel de Criticidad:	Media
Frecuencia de Ejecución:	Mensual, el 15 de cada mes.

**Fuente:** Propia

### Proceso 18: Seguimiento Diario de Asignación de Scores al TRIAD

Descripción: Validar que todos los scores lleguen al triad correctamente. La segunda parte validar que se ejecuten correctamente los scores. La salida son tablas SAS y se exporta a Excel. También hay salidas en PDF.

**Tabla N° 32:** Tabla Seguimiento\_diario

Responsable de Ejecución:	Paolo Ugarte
Usuario Final de datos generados:	Estrategia
Área:	Cobranzas
Nivel de Criticidad:	Alta
Frecuencia de Ejecución:	Diaria

Fuente: Propia

### Proceso 19: Cálculos Scores CBM

Descripción: Validar la correcta asignación de scores a las cuentas que entran diariamente a cobranzas, comparando con el score final y la puntuación de cada una de las variables que lo componen. La salida son tablas SAS y se exporta a Excel. También hay salidas en PDF.

**Tabla N° 33:** Tabla Scores\_CBM

Responsable de Ejecución:	Paolo Ugarte
Usuario Final de datos generados:	Cobranzas y comité de modelos.
Área:	Cobranzas
Nivel de Criticidad:	Media
Frecuencia de Ejecución:	Diaria

Fuente: Propia

### Proceso 20: Tablero Control de calidad

Descripción: Procesar información y obtener indicadores para llenar el Tablero de Control de Calidad. Se actualizan tablas Oracle al final de cada flujo.

**Tabla N° 34:** Tabla Control\_de\_calidad

Responsable de Ejecución:	Rosario López
Usuario Final de datos generados:	RBM
Área:	Riesgo Banca Minorista
Nivel de Criticidad:	Alta
Frecuencia de Ejecución:	Diaria

Fuente: Propia



### **Proceso 21: Tablero Control banca negocio**

Descripción: Procesar información y obtener indicadores para llenar el Tablero de Control de Calidad. Se actualizan tablas Oracle al final de cada flujo.

**Tabla N° 35:** Tabla Control\_banca\_negocio

Responsable de Ejecución:	Rosario López
Usuario Final de datos generados:	RBM
Área:	Riesgo Banca Minorista
Nivel de Criticidad:	Alta
Frecuencia de Ejecución:	Diaria

Fuente: Propia

### **Proceso 22: Tablero APP pyme**

Descripción: Procesar información y obtener indicadores para llenar el Tablero de Control de Calidad. Se actualizan tablas Oracle al final de cada flujo.

**Tabla N° 36:** Tabla Tablero\_APP\_pyme

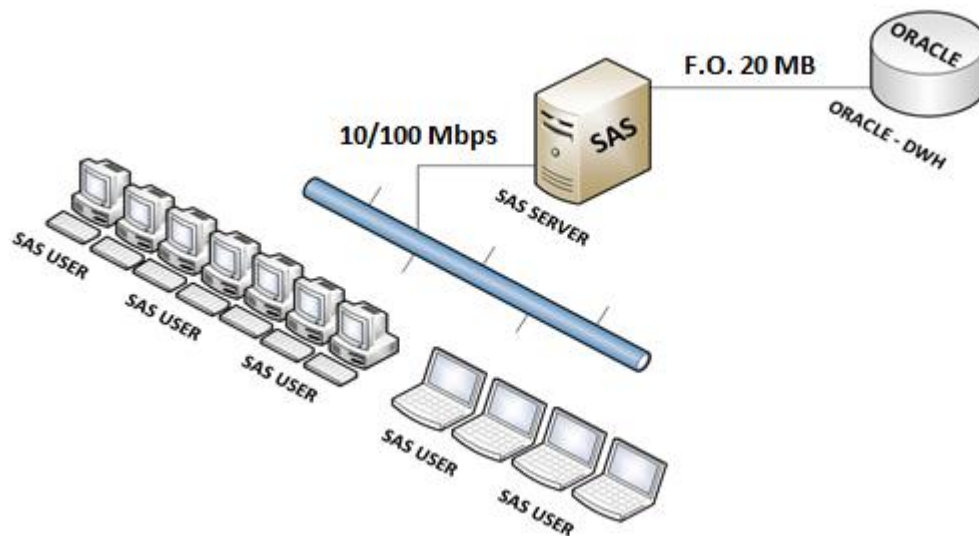
Responsable de Ejecución:	Rosario López
Usuario Final de datos generados:	RBM
Área:	Riesgo Banca Minorista
Nivel de Criticidad:	Alta
Frecuencia de Ejecución:	Diaria

Fuente: Propia

### **Salidas:**

- **Diagrama de Infraestructura Actual**

De acuerdo a la información brindada por el área de Sistemas, se realizó el diseño de la Infraestructura Centralizada que es como se trabaja actualmente.

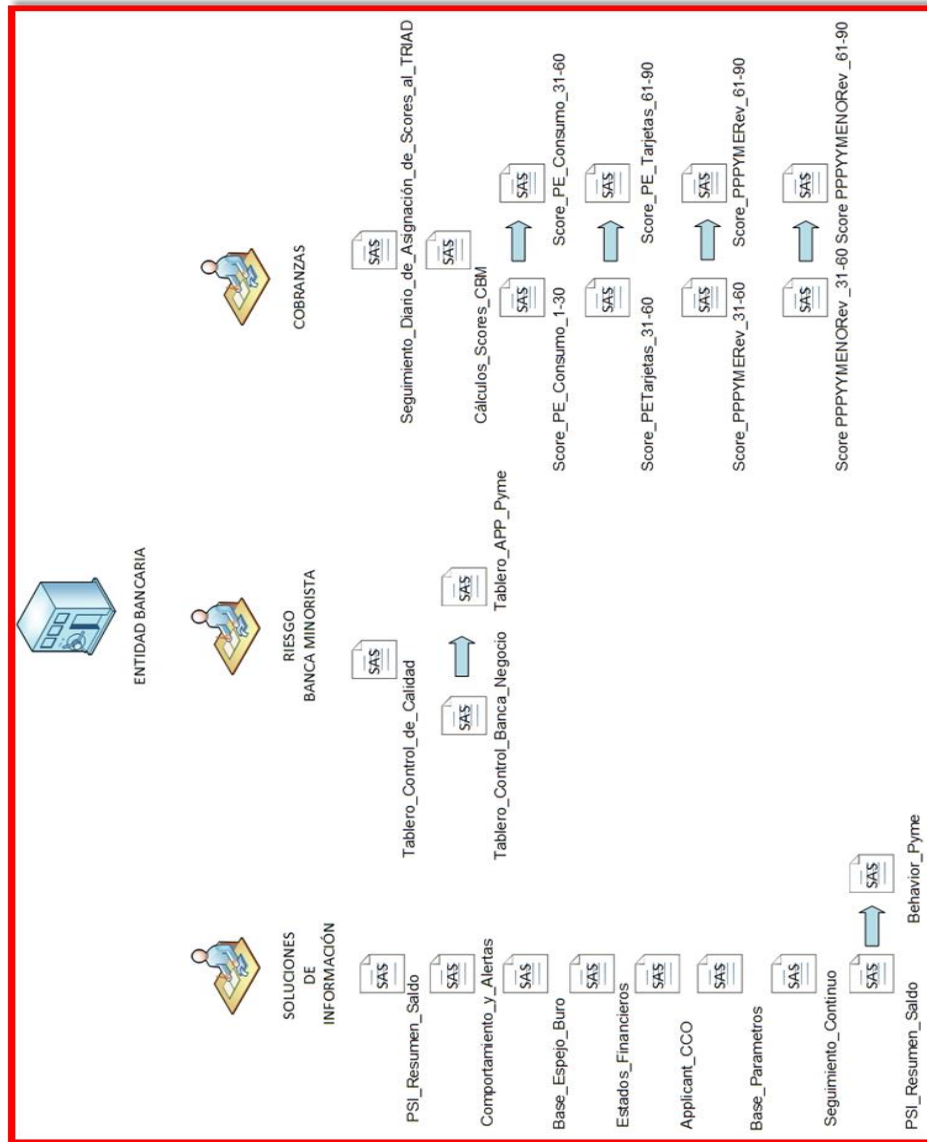


**Figura N° 11:** Diagrama de la Infraestructura Actual  
**Fuente:** Propia

El diagrama muestra, un servidor SAS que se encuentra alojado bajo la plataforma Windows Server 2008 R2, el cual es el intermediario entre el datawarehouse con servidor Oracle y los usuarios SAS que son los encargados de mandar a ejecutar los procesos de las diferentes áreas de la entidad bancaria.

- **Diagrama de procesos SAS**

La entidad bancaria ejecuta actualmente 22 procesos SAS en el servidor, estos procesos provienen de diferentes áreas que se detallan en el siguiente diagrama:



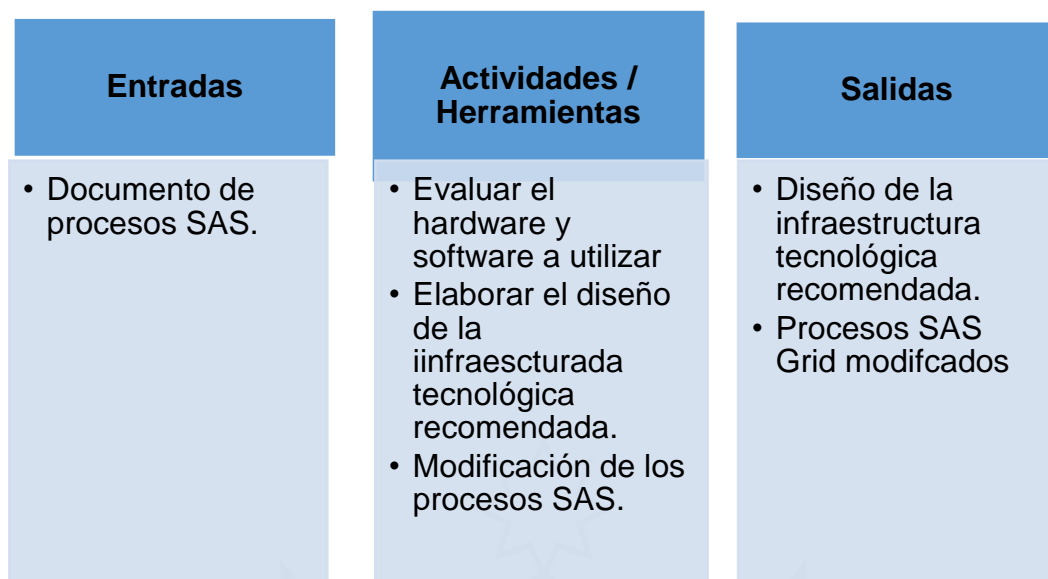
**Figura N° 12:** Diagrama de los procesos SAS de la entidad bancaria  
Fuente: Propia

### 3.3 Diseño de la infraestructura propuesta SAS Grid

#### Fase III: Diseño del sistema recomendado

De acuerdo, a nuestra metodología a seguir Kendall & Kendall, para la fase III se seguirán lo siguientes lineamientos:

**Tabla N° 37:** Diagrama de Entradas, Actividades/Herramientas, Salidas de diseño de solución



Fuente: Propia

#### Documentos de Entrada:

Como documento de entrada tenemos el documento de procesos SAS, del cual solo se tomará los nueve procesos del área de soluciones de información que pasarán unas ciertas modificaciones para así puedan ser ejecutados bajo la nueva plataforma recomendada.

#### Actividades:

- **Evaluar el hardware y software a utilizar**

Debido al alto nivel de procesamiento que consumen los procesos SAS de la entidad bancaria se determina usar lo siguiente:

**Tabla N° 38:** Tabla de evaluación de hardware y software a utilizar

Servidor	Hardware	Software
Metadata	<ul style="list-style-type: none"> <li>✓ CPU de 1 core</li> <li>✓ Memoria de 8 GB RAM</li> <li>✓ 160 GB de Disco</li> </ul>	<ul style="list-style-type: none"> <li>✓ SO IBM AIX</li> <li>✓ Versión SAS 9.4</li> </ul>
Mid Tier	<ul style="list-style-type: none"> <li>✓ CPU de 1 core</li> <li>✓ Memoria de 8 GB RAM</li> <li>✓ 160 GB de Disco</li> </ul>	<ul style="list-style-type: none"> <li>✓ SO IBM AIX</li> <li>✓ Versión SAS 9.4</li> </ul>
Grid Manager	<ul style="list-style-type: none"> <li>✓ CPU de 2 core</li> </ul>	<ul style="list-style-type: none"> <li>✓ SO IBM AIX</li> </ul>

	<ul style="list-style-type: none"> <li>✓ Memoria de 8 GB RAM</li> <li>✓ 160 GB de Disco</li> </ul>	<ul style="list-style-type: none"> <li>✓ Versión SAS 9.4</li> </ul>
Fail Over	<ul style="list-style-type: none"> <li>✓ CPU de 2 core</li> <li>✓ Memoria de 8 GB RAM</li> <li>✓ 160 GB de Disco</li> </ul>	<ul style="list-style-type: none"> <li>✓ SO IBM AIX</li> <li>✓ Versión SAS 9.4</li> </ul>
Grid Node	<ul style="list-style-type: none"> <li>✓ CPU de 2 core</li> <li>✓ Memoria de 8 GB RAM</li> <li>✓ 160 GB de Disco</li> </ul>	<ul style="list-style-type: none"> <li>✓ SO IBM AIX</li> <li>✓ Versión SAS 9.4</li> <li>✓ SAS/Connect 9.4</li> <li>✓ SAS/STAT 9.4</li> </ul>

Fuente: Propia

### Conectividad SAN (Storage area network)

Red de almacenamiento integral. Se tratará de una arquitectura completa que agrupará una red de alta velocidad de canal de fibra, unos equipos de interconexión dedicados como conmutadores y puentes y elementos de almacenamiento de red como discos duros. La red SAN tiene una interfaz de red específica que se conecta a la SAN.

El rendimiento de la SAN está directamente relacionado con el tipo de red que se utiliza. En el diseño recomendado se usará una red de canal de fibra, el ancho de banda es de aproximadamente 100 megabytes/segundo (1.000 megabits/segundo) y se puede extender aumentando la cantidad de conexiones de acceso. La capacidad de una SAN se puede extender de manera casi ilimitada y puede alcanzar cientos y hasta miles de terabytes.

La SAN nos permitirá compartir datos entre varios equipos de la red sin afectar el rendimiento porque el tráfico de SAN está totalmente separado del tráfico de usuario. Son los servidores de aplicaciones que funcionan como una interfaz entre la red de datos canal de fibra y la red de usuario Ethernet.

Se usará para el diseño recomendado las siguientes especificaciones:

- Sistema de almacenamiento IBM SAB24B.
- Fibra óptica (1 switch).
- 8 adaptadores para conectar los nodos de grid.
- Los puertos del switch SAN correrán en modo autonegociación (velocidad del puerto en Gbps).

Se consideró para el diseño de la solución cuatro nodos grid debido a los 22 procesos SAS que se ejecutan actualmente, los cuales se repartirán la carga de procesamiento para lograr disminuir los tiempos de ejecución de los procesos SAS.

Con esto se busca ejecutar los procesos SAS para aumentar la disponibilidad del servidor, mejorar los tiempos de ejecución, y así finalmente se obtengan los reportes analíticos a tiempo para la toma de decisiones gerenciales del área de soluciones de información.

- **Elaborar el diseño de la infraestructura tecnológica recomendada**

Basándonos en la teoría de grid computing se deben considerar tres capas fundamentales:

- Capa de acceso
- Capa de servicio
- Capa de recursos

Para la capa de acceso, se tendrá el servidor mid tier, el cual es un servidor de aplicaciones que interactúa directamente con el usuario.

Para la capa de servicio, el servidor metadata, es donde se encuentra la seguridad del SAS.

Para la capa de recursos, el servidor Grid Manager es quien se encarga de distribuir la carga entre los nodos Grid.

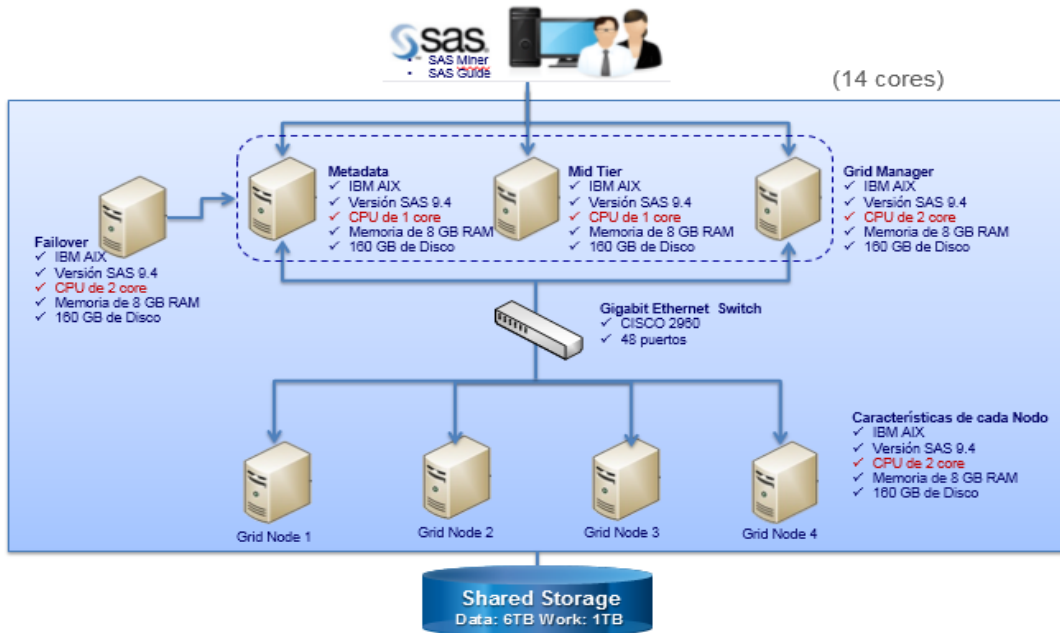
## **Descripción de la Infraestructura recomendada**

Se describirán los servidores necesarios para el diseño de la infraestructura recomendada que son los siguientes:

- Servidor Failover: Es el backup del servidor Metadata, ante alguna caída.
- Servidor Metadata: Este servidor contiene el repositorio de metadatos, la seguridad de SAS para usuarios, grupos, roles con sus respectivos permisos, estructuras para organizar los distintos objetos en Folders de metadata y una serie de objetos según cada herramienta que se tenga instalada como proyectos miner, proyectos guide, stored procedures, etc.
- Servidor Mid Tier: Es el servidor de aplicaciones web. Nos permite realizar el procesamiento de SAS como consultas y análisis de datos.
- Servidor Grid Manager: Es el servidor que se encarga de distribuir los procesos entre los demás nodos (nodo 1, 2 3 y 4) y los distribuye según algoritmos internos de SAS, según las reglas que se definan y según la disponibilidad de cada nodo.
- Shared Storage: El almacenamiento compartido es donde se encuentran todos los discos duros y de donde leen y escriben los nodos, hace uso de la arquitectura de almacenamiento SAN (Red de área de almacenamiento).

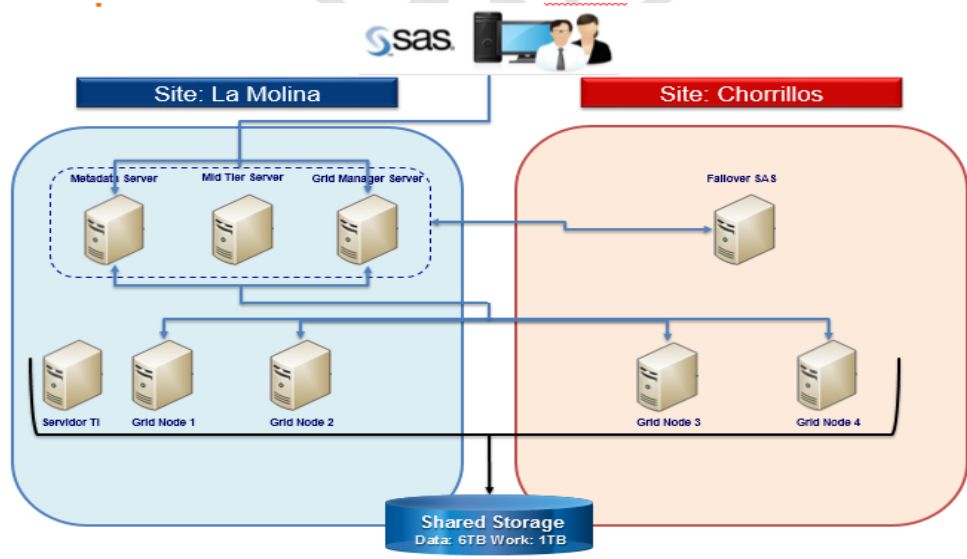
## Diseño de la infraestructura recomendada

Se procede a realizar el Diseño de Infraestructura Recomendado para la entidad bancaria.



**Figura N° 13:** Diseño de la Infraestructura Tecnológica Recomendada  
**Fuente:** Propia

De la solución recomendada, se estableció que en la sede Chorrillos se implementaran dos servidores grid y el servidor backup Failover y en sede La Molina quedará con los servidores Metadada, Mid Tier, Grid Manager y dos servidores Grid.



**Figura N° 14:** Distribución de los servidores Grid  
**Fuente:** Propia



### 3.4 Realización de la migración de procesos SAS actuales del área de Soluciones de Información para la nueva infraestructura SAS Grid

#### FASE III: Diseño del sistema recomendado

#### Documentos de Entrada:

Como documento de entrada tenemos el documento de procesos SAS, del cual solo se tomará los nueve procesos del área de soluciones de información que pasarán unas ciertas modificaciones para así puedan ser ejecutados bajo la nueva plataforma recomendada.

#### Actividades:

Para que los procesos SAS puedan ser ejecutados en la nueva plataforma, deberán pasar por un plan de migración que será de la siguiente manera:

#### Pasos para elaborar plan de migración

Todos los programas que componen los procesos del área de soluciones de información, deberán tener añadidos las siguientes líneas al inicio:

#### Activación servicio Grid:

```
%let rc=%sysfunc(grdsvc_enable(_all_,server=SASApp));  
options autosignon;
```

Variables adicionales para ejecución Jobs Grid:

```
options emailhost=("172.29.64.20" port=25); → Ruta del servidor grid
```

```
%let mailaddr= S40073@CREDITO.BCP.COM.PE; → Correo notificación de  
ejecución del proceso al usuario.
```

#### Comienzo del Job:

Al inicio del programa, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/
```

```
%syslput FilePref=&FilePref /remote=job_x; → Se llamarán a las variables  
que se utilizarán dentro del proceso
```

***/\*Inicializar fecha de comienzo del job\*/***

```
data _null_;  
    tempdate=put(date(),ddmmyyn8.);  
    temptime=put(time(),b8601tm.);  
    call symput('currdate',catx('_',tempdate,temptime));  
run;
```

***/\*Definición del Job\*/***

```
rsubmit job_asg wait=no cmacvar=job_asg sysrputsync=yes inheritlib=(data)  
→ Librerías a llamar durante la ejecución del proceso  
log="/sasdata01/Soluciones_Informacion/Indice_Estabilidad/logs/Ejecucion_j  
ob_ASG_&currdate..log" new→ruta del log del proceso  
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio  
Ejecucion Job_ASG ";
```

```
options fullstimer mprint mlogic;
```

...

**Fin del job:**

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

**Ejecución del job:**

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_asg;
```

La planificación detallada de los procesos SAS se encuentran en el anexo 3.

## CAPÍTULO IV

### PRUEBAS Y RESULTADOS

#### 4.1 Pruebas

Para las pruebas de simulación del entorno SAS Grid se utilizó lo siguiente:

- **Servidor físico de prueba utilizado para virtualizar el entorno SAS Grid**

Procesador: Core i7 6700k sexta generación 4GHz 4 nucleos

Memoria Ram: 64 GB

Disco Duro: 2.4 TB

Sistema Operativo: Windows 7

Software para Virtualizar: VMWARE 12

- **Servidores Virtuales**

Memoria Ram: 2 GB

Disco Duro: 160 GB

Sistema Operativo: IBM AIX 7.1

#### 4.2 Resultados

Como resultados, se muestra la mejora en los tiempos de ejecución de los procesos SAS del área de soluciones de información, en base a los logs de ejecución de la infraestructura SAS con la ejecución en un servidor de pruebas para SAS Grid.

## Proceso Comportamiento\_y\_alertas

**Tabla N° 39:** Tiempo de ejecución proceso Comportamiento\_y\_alertas del área de Soluciones de Información (HH:MM:SS)

			SERVIDOR ACTUAL	SIMULACIÓN SERVIDOR SAS GRID	DISMINUCIÓN DE TIEMPO
Tiempo total de ejecución	Log	Tiempo Total	08:37:00	02:42:00	05:55:00
		Tiempo job_Importacion	N/A	01:23:00	00:00:00
		Tiempo job_TablaUnica	N/A	00:32:12	00:00:00
		Tiempo job_Modelo	N/A	00:28:07	00:00:00

Fuente: Propia

Conclusión: El proceso SAS Comportamiento\_y\_alertas del área de Soluciones de Información tarda actualmente en ejecutarse 8 horas con 37 minutos, en la simulación para la plataforma sas grid tardó 2 horas con 42 minutos observando una mejora de 5 horas y 45 minutos en la ejecución del proceso, esto permite tener como resultado final la entrega del reporte en la fecha establecida evitando el retraso en la entrega del reporte y la toma de decisiones de la gerencia.

## Proceso Base\_Espejo\_Buro

**Tabla N° 40:** Tiempo de ejecución proceso Base\_espejo\_buro del área de Soluciones de Información (HH:MM:SS)

			SERVIDOR ACTUAL	SIMULACIÓN SERVIDOR SAS GRID	DISMINUCIÓN DE TIEMPO
Tiempo total de ejecución	Log	Tiempo Total	07:15:00	03:00:12	04:14:48
		Tiempo Soluciones_Información_job_djf	N/A	02:08:00	00:00:00
		Tiempo Soluciones_Información_job_scr	N/A	01:04:00	00:00:00

Fuente: Propia

Conclusión: El proceso SAS Base\_espejo\_buro del área de Soluciones de Información tarda actualmente en ejecutarse 7 horas con 15 minutos, en la simulación para la plataforma sas grid tardó 3 horas con 12 minutos observando una mejora de 4 horas y 14 minutos en la ejecución del proceso,

esto permite tener como resultado final la entrega del reporte en la fecha establecida evitando el retraso en la entrega del reporte y la toma decisiones de la gerencia.

### Proceso PSI\_resumen\_saldo

**Tabla N° 41:** Tiempo de ejecución proceso PSI\_resumen\_saldo del área de Soluciones de Información (HH:MM:SS)

			SERVIDOR ACTUAL	SIMULACIÓN SERVIDOR SAS GRID	DISMINUCIÓN DE TIEMPO
Tiempo total de ejecución	Log	Tiempo Total	12:25:00	05:38:00	06:37:00
		Tiempo Asignacion_job	N/A	03:16:00	00:00:00
		Tiempo Compare_job	N/A	02:12:00	00:00:00

Fuente: Propia

Conclusión: El proceso SAS PSI\_resumen\_saldo del área de Soluciones de Información tarda actualmente en ejecutarse 12 horas con 25 minutos, en la simulación para la plataforma sas grid tardó 5 horas con 38 minutos observando una mejora de 6 horas y 37 minutos en la ejecución del proceso, esto permite tener como resultado final la entrega del reporte en la fecha establecida evitando el retraso en la entrega del reporte y la toma decisiones de la gerencia.

### Proceso Applicant\_cco

**Tabla N° 42:** Tiempo de ejecución proceso Applicant\_cco del área de Soluciones de Información (HH:MM:SS)

			SERVIDOR ACTUAL	SIMULACIÓN SERVIDOR SAS GRID	DISMINUCIÓN DE TIEMPO
Tiempo total de ejecución	Log	Tiempo Total	05:27:00	02:52:00	02:35:00
		Tiempo job_Des	N/A	01:10:00	00:00:00
		Tiempo job_Calc	N/A	00:50:00	00:00:00
		Tiempo job_Carg	N/A	00:52:00	00:00:00

Fuente: Propia

Conclusión: El proceso SAS Applicant\_cco del área de Soluciones de Información tarda actualmente en ejecutarse 5 horas con 27 minutos, en la simulación para la plataforma sas grid tardó 2 horas con 52 minutos observando una mejora de 2 horas y 35 minutos en la ejecución del proceso, esto permite tener como resultado final la entrega del reporte en la fecha establecida evitando el retraso en la entrega del reporte y la toma decisiones de la gerencia.

### Proceso Estados\_financieros

**Tabla N° 43:** Tiempo de ejecución proceso Estados\_financieros del área de Soluciones de Información (HH:MM:SS)

			SERVIDOR ACTUAL	SIMULACIÓN SERVIDOR SAS GRID	DISMINUCIÓN DE TIEMPO
Tiempo total de ejecución	Log	Tiempo Total	04:10:00	02:42:00	01:28:00
		Tiempo job_Inc	N/A	01:13:00	00:00:00
		Tiempo job_Cons	N/A	00:35:00	00:00:00
		Tiempo job_tot	N/A	00:54:00	00:00:00

Fuente: Propia

Conclusión: El proceso SAS Estados\_financieros del área de Soluciones de Información tarda actualmente en ejecutarse 4 horas con 10 minutos, en la simulación para la plataforma sas grid tardó 2 horas con 42 minutos observando una mejora de 1 hora y 28 minutos en la ejecución del proceso, esto permite tener como resultado final la entrega del reporte en la fecha establecida evitando el retraso en la entrega del reporte y la toma decisiones de la gerencia.

### Procesos Applicant\_pyme

**Tabla N° 44:** Tiempo de ejecución proceso Applicant\_pyme del área de Soluciones de Información (HH:MM:SS)

			SERVIDOR ACTUAL	SIMULACIÓN SERVIDOR SAS GRID	DISMINUCIÓN DE TIEMPO
Tiempo total de ejecución	Log	Tiempo Total	06:23:00	02:56:00	03:27:00
		Tiempo job_Desc	N/A	00:52:10	00:00:00

	Tiempo job_Calc	N/A	00:22:20	00:00:00
	Tiempo job_Carg	N/A	00:28:05	00:00:00
	Tiempo job_Appn	N/A	00:13:07	00:00:00
	Tiempo job_Appre	N/A	00:07:02	00:00:00

Fuente: Propia

Conclusión: El proceso SAS Applicant\_pyme del área de Soluciones de Información tarda actualmente en ejecutarse 6 horas con 23 minutos, en la simulación para la plataforma sas grid tardó 2 horas con 56 minutos observando una mejora de 3 horas y 27 minutos en la ejecución del proceso, esto permite tener como resultado final la entrega del reporte en la fecha establecida evitando el retraso en la entrega del reporte y la toma decisiones de la gerencia.

### Proceso Base\_parámetros

**Tabla N° 45:** Tiempo de ejecución proceso Base\_parámetros del área de Soluciones de Información (HH:MM:SS)

		SERVIDOR ACTUAL	SIMULACIÓN SERVIDOR SAS GRID	DISMINUCIÓN DE TIEMPO	
Tiempo total de ejecución	Log	Tiempo Total	06:52:00	02:15:00	04:37:00
		MinoristaTarjetas_job	N/A	00:30:12	00:00:00
		MinoristaPymRev_job	N/A	00:20:09	00:00:00
		MinoristaPymNoRev_job	N/A	00:22:10	00:00:00
		MinoristaHipotecario_job	N/A	00:15:48	00:00:00
		MinoristaConsumo_job	N/A	00:10:02	00:00:00
		MinoristaVehicular_job	N/A	00:23:08	00:00:00

Fuente: Propia

Conclusión: El proceso SAS Base\_parámetros del área de Soluciones de Información tarda actualmente en ejecutarse 6 horas con 52 minutos, en la simulación para la plataforma sas grid tardó 2 horas con 15 minutos observando una mejora de 4 horas y 37 minutos en la ejecución del proceso, esto permite tener como resultado final la entrega del reporte en la fecha

establecida evitando el retraso en la entrega del reporte y la toma decisiones de la gerencia.

### Proceso Seguimiento\_continuo

**Tabla N° 46:** Tiempo de ejecución proceso Seguimiento\_continuo del área de Soluciones de Información (HH:MM:SS)

			SERVIDOR ACTUAL	SIMULACIÓN SERVIDOR SAS GRID	DISMINUCIÓN DE TIEMPO
Tiempo total de ejecución	Log	Tiempo Total	09:40:00	03:40:00	06:00:00
		Tiempo Consumo_job_for m_&currdate..log	N/A	00:33:04	00:00:00
		Tiempo Consumo_job_pr ep_&currdate..log	N/A	00:27:09	00:00:00
		Tiempo Consumo_job_im p_&currdate..log	N/A	00:32:10	00:00:00
		Tiempo Consumo_job_Ca tV1_&currdate..lo g	N/A	00:22:05	00:00:00
		Tiempo Consumo_job_Ca tV2_&currdate..lo g	N/A	00:12:35	00:00:00
		Tiempo Consumo_job_Ca tV3_&currdate..lo g	N/A	00:15:07	00:00:00
		Tiempo Consumo_job_Du mmy_&currdate..l og	N/A	00:08:35	00:00:00
		Tiempo Consumo_job_TA dm_&currdate..lo g	N/A	00:36:17	00:00:00
		Tiempo Consumo_job_T Comp_&currdate..log	N/A	00:28:04	00:00:00

Fuente: Propia

Conclusión: El proceso SAS Seguimiento\_continuo del área de soluciones de información tarda actualmente en ejecutarse 9 horas con 40 minutos, en la simulación para la plataforma sas grid tardó 3 horas con 40 minutos observando una mejora de 6 horas y 0 minutos en la ejecución del proceso, esto permite tener como resultado final la entrega del reporte en la fecha



establecida evitando el retraso en la entrega del reporte y la toma decisiones de la gerencia.

### Proceso Behavior\_pyme

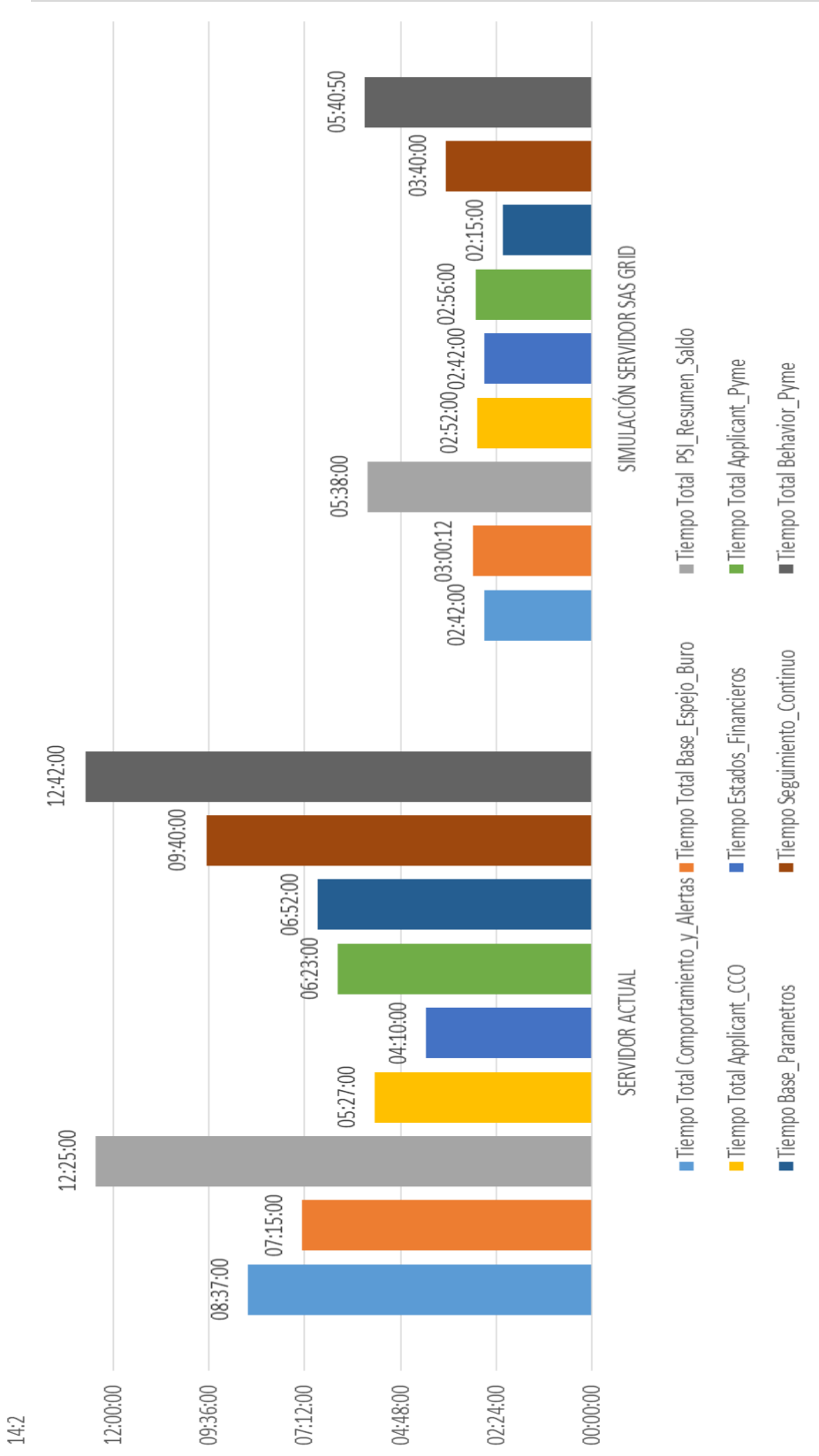
**Tabla N° 47:** Tiempo de ejecución proceso Behavior\_pyme del área de Soluciones de Información (HH:MM:SS)

		SERVIDOR ACTUAL	SIMULACIÓN SERVIDOR SAS GRID	DISMINUCIÓN DE TIEMPO	
Tiempo total de ejecución	Log	Tiempo Total	12:42:00	05:40:50	07:01:10
		Job Universo_Clientes_Pyme_job_UPyme_&currdate..log	N/A	01:05:04	00:00:00
		Job Modulo_Solicitudes_job_MSol_&currdate..log	N/A	01:12:05	00:00:00
		Job Modulo_Activos_job_MAct_&currdate..log	N/A	01:03:07	00:00:00
		Job Modulo_Pasivos_job_MPas_&currdate..log	N/A	01:08:30	00:00:00
		Job Modulo_RCC_job_MRCC_&currdate..log	N/A	01:12:04	00:00:00

Fuente: Propia

Conclusión: El proceso SAS Behavior\_pyme del área de soluciones de información tarda actualmente en ejecutarse 12 horas con 42 minutos, en la simulación para la plataforma sas grid tardó 5 horas con 40 minutos observando una mejora de 7 horas y 1 minuto en la ejecución del proceso, esto permite tener como resultado final la entrega del reporte en la fecha establecida evitando el retraso en la entrega del reporte y la toma decisiones de la gerencia.

**Tabla N° 48:** Tiempo de ejecución de los procesos SAS del área de Soluciones de Información



**Fuente:** Propia

**Tabla N° 49:** Porcentaje de disminución de la diferencia en min entre el tiempo en minutos servidor actual v tiempo en minutos en el simulador del servidor SAS GRID

PROCESOS SAS	TIEMPO EN MIN SERVIDOR ACTUAL	TIEMPO EN MIN SIMULACIÓN SERVIDOR SAS GRID	% DE DISMINUCIÓN DE LA DIFERENCIA EN MIN
Tiempo Total Comportamiento_y_Alertas	517 MIN	355 MIN	68.67 %
Tiempo Total Base_Espejo_Buro	435 MIN	254 MIN	49.13 %
Tiempo Total PSI_Resumen_Saldo	745 MIN	407 MIN	78.72 %
Tiempo Total Applicant_CCO	327 MIN	155 MIN	29.98 %
Tiempo Estados_Financieros	250 MIN	88 MIN	17.02 %
Tiempo Total Applicant_Pyme	383 MIN	207 MIN	40.04 %
Tiempo Base_Parámetros	412 MIN	277 MIN	53.58 %
Tiempo Seguimiento_Continuo	580 MIN	360 MIN	69.63 %
Tiempo Total Behavior_Pyme	762 MIN	421 MIN	81.43 %

Fuente: Propia

## **CAPÍTULO V**

### **DISCUSIÓN Y APLICACIÓN**

#### **5.1 Discusión**

La Entidad Bancaria cuenta con 22 procesos SAS que son los encargados de emitir reportes analíticos sumamente importantes para la toma de decisiones gerenciales de la entidad bancaria.

Debido a la gran cantidad de procesos enviados de manera simultánea a un único servidor, el cual es centralizado, ocasiona que los procesos no se ejecuten completamente generando una demora en la entrega de los reportes y ocasionando pérdidas de dinero.

#### **5.2 Aplicación**

Como aplicación para la entidad bancaria es favorable implementar la infraestructura SAS Grid para que apoye al soporte de ejecución de los procesos SAS, dado que se evitará el retraso en la toma de decisiones gerenciales y pérdidas monetarias.



## CONCLUSIONES

- Primera:** La simulación de la infraestructura SAS Grid logró optimizar la ejecución de los procesos SAS para el soporte a la toma de decisiones gerenciales del área de Soluciones de Información de una entidad bancaria.
- Segunda:** El análisis del documento de incidencias permitió identificar los motivos y el problema que presentaba la infraestructura.
- Tercera:** La simulación de la infraestructura SAS Grid logró reducir los tiempos de ejecución de los procesos SAS del área de Soluciones de Información en aproximadamente 54.24 %, lo cual permitió tener los reportes analíticos para la entidad bancaria en la fecha establecida.
- Cuarta:** El diseño de una infraestructura eficiente permitió apoyar la toma de decisiones gerenciales para el cumplimiento de los objetivos estratégicos del área de Soluciones de Información de la entidad bancaria.
- Quinta:** La migración de los procesos SAS del área de Soluciones de Información se realizó para su ejecución en la nueva infraestructura SAS Grid bajo la plataforma AIX.

## RECOMENDACIONES

- Primera:** La entidad bancaria debería implementar la infraestructura Grid para lograr optimizar los tiempos de ejecución de los procesos SAS, aumentar la disponibilidad y el balanceo de carga.
- Segunda:** La entidad bancaria debería realizar la migración de todos los procesos SAS de las diferentes áreas.



## FUENTES DE INFORMACIÓN

- Amaya, J. (2010). *Toma de decisiones gerenciales. Métodos cuantitativos para la administración*. México: Ediciones ECOE.
- BCP (2013). *Historia del Banco de Crédito*. Recuperado de <https://www.viabcp.com/wps/portal/viabcpp/nuestro-banco/quienes-somos/historia>
- Calvo, A., Parejo, J., Rodríguez, L. & Cuervo, A. (2010). *Manual del sistema financiero español*. España: Ediciones ARIEL.
- Chugnas, J. (2014). *Sistema Financiero Peruano*. Recuperado de <http://es.slideshare.net/carlovmartinezblog/sesin-1-la-funcin-financiera-el-sistema-financiero-peruano>
- Castro, H. (2006). *Grid Computing: promesa de los sistemas distribuidos*. Colombia: Revista Sistemas ACIS.
- Chugnas Jonathan. (2011). *Sistema Bancario en el Perú*. Cajamarca: Universidad Nacional de Cajamarca.
- Delgado & Lozano. (2014). *Sistema para el control de ingreso de estudiantes para la conformación del equipo juvenil de básquetbol. Vicerrectorado de planificación y desarrollo social (VPDS) del programa de arquitectura, ingeniería y tecnología, del subprograma de Ingeniería Informática 2014*. Venezuela: Universidad Nacional Experimental de los Llanos Occidentales.
- Enriquez, S. & Ysaías, H. (2007). *Clustering y Grid Computing*. Trujillo: Escuela de Informática. Universidad Nacional de Trujillo.
- Gómez. (2014). *Grid y Clustering. Desarrollo de software de Altas Prestaciones*. Argentina: Ciencias Informáticas UNLP

Isaza, G. & Mendez, N. (2007). *Arquitecturas y modelos de programación en computación grid*. Colombia: Universidad Tecnológica de Colombia.

Kendall & Kendall. (2011). *Análisis y Diseño de Sistemas*. México: Pearson Educación.

Pérez, M. (2013). *Sistema de Análisis Estadístico SAS. Lenguaje de Programación*. Madrid: Ediciones RC

SAS Institute. (2015). *Grid Computing in SAS 9.4*. Estados Unidos: SAS Institute.

Senn, J. (1996). *Análisis y Diseño de sistemas*. México: Ediciones Mc Graw Hill.







**ANEXOS**

## ANEXO 1 INCIDENTES SAS 2016

Se coloca el archivo de Incidencias\_SAS\_2016 otorgado por el área de Sistemas de la entidad bancaria

**Tabla:** Registro de Incidencias Enero 2016

N°	Proceso SAS	Area de Proceso	Motivo	Tiempo de ejecución	Fecha Ocurrencia	Usuario Solicitante
1	Seguimiento Continuo	Soluciones de Información	Proceso cayó al ejecutar quinto flujo	10 h 29 min	04/01/2016	Boris Sinche
2	Proyecto Modelos de propención para seguros	Riesgo Banca Minorista	Proceso cayó al ejecutar cuarto flujo	8 h 45 min	04/01/2016	Rosario Gonzáles
3	Seguimiento Continuo	Soluciones de Información	Proceso cayó al ejecutar tercer flujo	8 h 52 min	07/01/2016	Ivan Mendoza
4	Behavior Pyme	Soluciones de Información	Proceso cayó al ejecutar cuarto flujo	7 h 35 min	07/01/2016	Boris Sinche
5	Tablero Control Calidad	Riesgo Banca Minorista	Proceso cayó al ejecutar tercer flujo	8 h 25 min	11/01/2016	Rosario Gonzáles
6	Behavior Pyme	Soluciones de Información	Proceso cayó al ejecutar tercer flujo	10 h 20 min	11/01/2016	Boris Sinche
7	Seguimiento Continuo	Soluciones de Información	Proceso cayó al ejecutar cuarto flujo	11 h 50 min	13/01/2016	Ruddy Sánchez
8	Estados Financieros	Soluciones de Información	Proceso cayó por multisesión al llamar una tabla histórica	10 h 47 min	13/01/2016	Ivan Mendoza
9	Debito Personas	Riesgo Banca Minorista	Insuficiente espacio en el servidor psasp04	13 h 45 min	14/01/2016	Rosario Gonzáles
10	Saldos Mov. Fact. Vplus	Soluciones de Información	Proceso cayó por multisesión al llamar una tabla histórica	9 h 47 min	26/01/2016	Jimmy Farfan
11	Saldos Mov. Fact. Vplus	Soluciones de Información	Proceso cayó al ejecutar tercer flujo	8 h 15 min	28/01/2016	Jimmy Farfan
12	Seguimiento Continuo	Soluciones de Información	Proceso cayó al ejecutar segundo flujo	9 h 29 min	15/01/2016	Ruddy Caja
13	Estimadores de NBA y Combinado	Riesgo Banca Minorista	Proceso cayó al ejecutar cuarto flujo	8 h 26 min	18/01/2016	Luis Palacios
14	Grupos de Vinculación TC y CE	Riesgo Banca Minorista	Proceso cayó al ejecutar quinto flujo	9 h 45 min	18/01/2016	Rosario Gonzáles
15	Seguimiento Estadístico Score CBM	Cobranzas	Proceso cayó al ejecutar tercer flujo	8 h 29 min	20/01/2016	Mabel Gómez
16	Resumen Saldo	Soluciones de Información	Proceso cayó al ejecutar tercer flujo	7 h 52 min	22/01/2016	Jimmy Farfan
17	06_Base_EspejoBuro	Soluciones de Información	Proceso cayó al ejecutar cuarto flujo	7 h 45 min	22/01/2016	Fiorella Tenorio

Fuente: Propia

**Tabla:** Registro de Incidencias Febrero 2016

N°	Proceso SAS	Area de Proceso	Motivo	Tiempo de ejecución	Fecha Ocurrencia	Usuario Solicitante
1	Seguimiento Estadístico Score CBM	Cobranzas	Insuficiente espacio en el servidor psasp04	8 h 47 min	01/02/2016	Mabel Gómez
2	Tablero App Pyme	Riesgo Banca Minorista	Insuficiente espacio en el servidor psasp04	12 h 47 min	03/02/2016	Rosario Gonzáles
3	Seg. Diario de Asig. Score al TRIAD	Cobranzas	Proceso cayó por multisesión al llamar una tabla histórica	9 h 47 min	09/02/2016	Ruddy Sánchez
4	Sombra Hipotecario	Riesgo Banca Minorista	Proceso cayó por multisesión al llamar una tabla histórica	12h 25 min	09/02/2016	Luis Guillén
5	Seguimiento Continuo	Riesgo Banca Minorista	Insuficiente espacio en el servidor psasp04	13 h 45 min	12/02/2016	Rosario Gonzáles
6	Base Parametros	Soluciones de Información	Proceso cayó por multisesión al llamar una tabla histórica	9 h 26 min	26/02/2016	Ricardo Mimi
7	Tablero Control Calidad	Riesgo Banca Minorista	Proceso cayó por multisesión al llamar una tabla histórica	8 h 25 min	26/02/2016	Rosario Gonzáles
8	Estimadores de NBA y Combinado	Riesgo Banca Minorista	Proceso cayó al ejecutar segundo flujo	8 h 26 min	15/02/2016	Luis Palacios
9	Behavior Pyme	Soluciones de Información	Proceso cayó al ejecutar quinto flujo	10 h 29 min	25/02/2016	Boris Sinche
10	Seguimiento Continuo	Soluciones de Información	Proceso cayó al ejecutar tercer flujo	15 h 28 min	29/02/2016	Fiorella Tenorio
11	Grupos de Vinculación TC y CE	Riesgo Banca Minorista	Proceso cayó por multisesión al llamar una tabla histórica	9 h 45 min	18/02/2016	Rosario Gonzáles
12	Tablero Banca Negocio	Riesgo Banca Minorista	Proceso cayó al ejecutar quinto flujo	8 h 45 min	21/02/2016	Rosario Gonzáles
13	Base EspejoBuro	Soluciones de Información	Proceso cayó por multisesión al llamar una tabla histórica	7 h 45 min	22/02/2016	Fiorella Tenorio
14	Ratios GyP	Riesgo Banca Minorista	Proceso cayó por multisesión al llamar una tabla histórica	10 h 50 min	22/02/2016	Ruddy Caja

Fuente: Propia

**Tabla:** Registro de Incidencias Marzo 2016

N°	Proceso SAS	Area de Proceso	Motivo	Tiempo de ejecución	Fecha Ocurrencia	Usuario Solicitante
1	Seguimiento Estadístico Score CBM	Cobranzas	Proceso cayó al ejecutar tercer flujo	7 h 45 min	01/03/2016	Mabel Gómez
2	PSI_Resumen_Saldo	Soluciones de Información	Proceso cayó al ejecutar cuarto flujo	8 h 26 min	01/03/2016	Jimmy Farfan
3	Base Espejo Buro	Soluciones de Información	Insuficiente espacio en el servidor psasp04	14 h 45 min	04/03/2016	Fiorella Tenorio
4	Tablero_Control_Calidad	Riesgo Banca Minorista	Proceso cayó al ejecutar cuarto flujo	9 h 15 min	04/03/2016	Rosario López
5	Seguimiento Estadístico Score CBM	Cobranzas	Insuficiente espacio en el servidor psasp04	7 h 52 min	08/03/2016	Mabel Gómez
6	Seguimiento Continuo	Riesgo Banca Minorista	Proceso cayó al ejecutar tercer flujo	12h 25 min	12/03/2016	Luis Guillén
7	Estados Financieros	Soluciones de Información	Proceso cayó al ejecutar cuarto flujo	10 h 47 min	12/03/2016	Ivan Mendoza
8	Applicant Pyme	Soluciones de Información	Proceso cayó al ejecutar quinto flujo	7 h 52 min	19/03/2016	Jimmy Farfan
9	Behavior Pyme Rev	Soluciones de Información	Proceso cayó al ejecutar tercer flujo	10 h 29 min	22/03/2016	Boris Sinche
10	Seguimiento Continuo	Soluciones de Información	Proceso cayó al ejecutar cuarto flujo	13 h 45 min	25/03/2016	Jimmy Farfan
11	Sistema Financiero RCC	Soluciones de Información	Proceso cayó por multisesión al llamar una tabla histórica	9 h 29 min	25/03/2016	Jimmy Farfan
12	PSI Resumen Saldo	Soluciones de Información	Proceso cayó al ejecutar cuarto flujo	8 h 26 min	31/03/2016	Jimmy Farfan

Fuente: Propia

**Tabla: Registro de Incidencias Abril 2016**

Nº	Proceso SAS	Area de Proceso	Motivo	Tiempo de ejecución	Fecha Ocurrencia	Usuario Solicitante
1	Base_Espejo Buro	Soluciones de Información	Proceso cayó por multisesión al llamar una tabla histórica	7 h 45 min	01/04/2016	Fiorella Tenorio
2	Seguimiento Estadístico Score CBM	Cobranzas	Proceso cayó por multisesión al llamar una tabla histórica	7 h 45 min	01/04/2016	Mabel Gómez
3	PSI Resumen Saldo	Soluciones de Información	Proceso cayó al ejecutar cuarto flujo	8 h 26 min	08/04/2016	Jimmy Farfan
4	Seguimiento Estadístico Score CBM	Cobranzas	Proceso cayó al ejecutar quinto flujo	10 h 50 min	08/04/2016	Mabel Gómez
5	Seguimiento Continuo	Soluciones de Información	Proceso cayó al ejecutar tercer flujo	8 h 52 min	12/04/2016	Ivan Mendoza
6	Seguimiento Estadístico Score CBM	Cobranzas	Proceso cayó al ejecutar quinto flujo	9 h 29 min	15/04/2016	Mabel Gómez
7	Estimadores de NBA y Combinado	Riesgo Banca Minorista	Proceso cayó por multisesión al llamar una tabla histórica	8 h 26 min	18/04/2016	Luis Palacios
8	Seguimiento Continuo	Soluciones de Información	Proceso cayó por multisesión al llamar una tabla histórica	9 h 47 min	18/04/2016	Jimmy Farfan
9	Saldos Mov. Fact. Vplus	Soluciones de Información	Insuficiente espacio en el servidor psasp04	6 h 20 min	22/04/2016	Mabel Gómez
10	Estados Financieros	Soluciones de Información	Proceso cayó al ejecutar quinto flujo	8 h 15 min	25/04/2016	Jimmy Farfan
11	Seg. Diario de Asiq. Score al TRIAD - Calculos Scores C	Cobranzas	Proceso cayó al ejecutar tercer flujo	9 h 47 min	27/04/2016	Ruddy Sánchez
12	Debito Personas	Riesgo Banca Minorista	Insuficiente espacio en el servidor psasp04	13 h 45 min	27/04/2016	Rosario Gonzáles
13	Saldos Mov. Fact. Vplus	Soluciones de Información	Insuficiente espacio en el servidor psasp04	9 h 47 min	30/04/2016	Jimmy Farfan

Fuente: Propia

**Tabla: Registro de Incidencias Abril 2016**

Nº	Proceso SAS	Subproceso SAS	Area de Proceso	Motivo	Tiempo de ejecución	Fecha Ocurrencia	Usuario Solicitante
1	Base_Espejo.Buro		Soluciones de Información	Proceso cayó por multisesión al llamar una tabla histórica	7 h 45 min	02/05/2016	Fiorella Tenorio
2	Seguimiento Estadístico Score CBM	Score PE Consumo 1 30	Cobranzas	Proceso cayó por multisesión al llamar una tabla histórica	9 h 49 min	03/05/2016	Mabel Gómez
3	PSI Resumen Saldo		Soluciones de Información	Proceso cayó al ejecutar cuarto flujo	8 h 26 min	05/05/2016	Jimmy Farfan
4	Seguimiento Continuo		Soluciones de Información	Proceso cayó al ejecutar tercer flujo	12 h 50 min	09/05/2016	Fiorella Tenorio
5	Seguimiento Estadístico Score CBM	Score PPPYMERev (Tramo	Cobranzas	Proceso cayó al ejecutar quinto flujo	10 h 50 min	11/05/2016	Mabel Gómez
6	Comp y Alertas		Soluciones de Información	Proceso cayó al ejecutar tercer flujo	8 h 52 min	12/05/2016	Ivan Mendoza
7	Seguimiento Estadístico Score CBM	Score PPConsumo (Tramo	Cobranzas	Proceso cayó al ejecutar quinto flujo	9 h 29 min	16/05/2016	Mabel Gómez
8	Seguimiento Continuo		Soluciones de Información	Proceso cayó al ejecutar cuarto flujo	8 h 26 min	18/05/2016	Fiorella Tenorio
9	Sistema Financiero RCC		Soluciones de Información	Proceso cayó al ejecutar quinto flujo	9 h 47 min	18/05/2016	Jimmy Farfan
10	Seguimiento Estadístico Score CBM	Score PPPYMERev (Tramo	Cobranzas	Insuficiente espacio en el servidor psasp04	6 h 20 min	20/05/2016	Mabel Gómez
11	Saldos Mov. Fact. Vplus	Saldos Mov. Fact. Vplus 2	Soluciones de Información	Proceso cayó por multisesión al llamar una tabla histórica	8 h 15 min	23/05/2016	Jimmy Farfan
12	Seg. Diario de Asiq. Score al TRIAD - Calculos	Seguimiento Diario	Cobranzas	Proceso cayó al ejecutar tercer flujo	9 h 47 min	25/05/2016	Ruddy Sánchez
13	Seguimiento Continuo		Soluciones de Información	Proceso cayó al ejecutar cuarto flujo	13 h 45 min	25/05/2016	Rosario Gonzáles
14	Saldos Mov. Fact. Vplus	Saldos Mov. Fact. Vplus 1	Soluciones de Información	Insuficiente espacio en el servidor psasp04	9 h 47 min	26/05/2016	Jimmy Farfan

Fuente: Propia

## ANEXO 2 INFORME DE VIABILIDAD

### I. Descripción

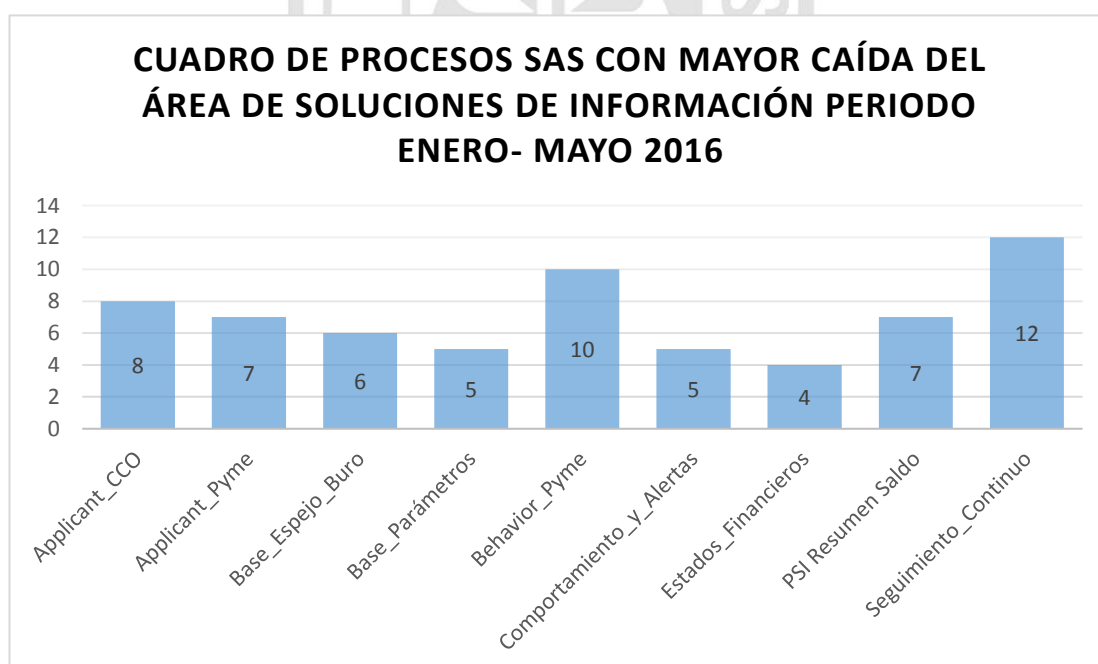
El presente documento tiene como finalidad sintetizar el problema, alternativas de solución y objetivo de este proyecto.

### II. Problema

Ineficiente infraestructura tecnológica que brinda soporte a la ejecución de los procesos SAS que ocasiona que los procesos no se ejecuten rápido y satisfactoriamente.

### III. Análisis de la información

Se analizó el archivo INCIDENTES\_SAS\_2016.xlsx otorgado por la entidad bancaria y se evidencia la cantidad de caídas ocurridas en los procesos del área de soluciones de información durante los meses de enero a mayo del presente año, que se detalla a continuación:

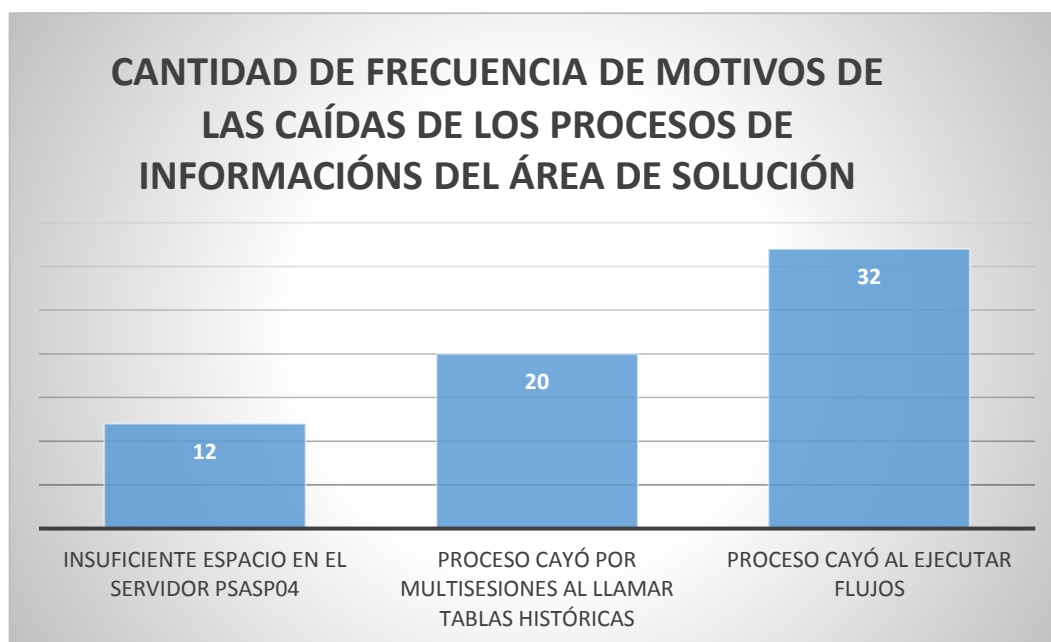


**Figura:** Cantidad de caídas por proceso del área de soluciones de información del presente año

**Fuente:** Propia

**Conclusión:** Mediante el gráfico, podemos observar que el proceso con mayor caída en el área de soluciones de información es el proceso de Seguimiento\_Continuo.

Los principales motivos que causan las caídas de los procesos del área de soluciones de información son los siguientes:



**Figura:** Cantidad de frecuencia de motivos de las caídas de los procesos del área de Soluciones de Información  
**Fuente:** Propia

Conclusión: Mediante el gráfico, podemos observar que la mayor concurrencia en los motivos de las caídas de los procesos del área de soluciones de información se debe a que el proceso cae al ejecutar los flujos de los programas sas.

Otra información destacada que se debe tomar en consideración, es la cantidad de procesos SAS por área y el peso que ocupan al realizar sus ejecuciones en el servidor sas PSASP04.

**Tabla:** Tabla de pesos en el servidor SAS PSASP04

Área	Procesos SAS	Peso en Disco
Soluciones de Información	9	8729 GB
Cobranzas	10	307 GB
Riesgo Banca Mayorista	7	1479 GB
Riesgo Banca Minorista	3	6019 GB

**Fuente:** Propia

También, se tomará en el análisis la información del tiempo de ejecución de los procesos SAS del área de soluciones de información:

**Tabla:** Tabla de Procesos del área de Soluciones de Información

Procesos SAS	Área	Tiempo de ejecución actual
Applicant_CCO	SINF	05:27:10 hh:mm:ss
Applicant_Pyme	SINF	06:23:32 hh:mm:ss
Base_Espejo_Buro	SINF	07:15:14 hh:mm:ss
Base_Parametros	SINF	06:52:50 hh:mm:ss
Behavior_Pyme	SINF	12:42:08 hh:mm:ss
Comportamiento_y_Alertas	SINF	08:37:24 hh:mm:ss
Estados_Financieros	SINF	04:10:35 hh:mm:ss
PSI Resumen Saldo	SINF	12:25:28 hh:mm:ss
Seguimiento_Continuo	SINF	09:40:45 hh:mm:ss

Fuente: Propia

#### IV. Alternativa de Solución

De acuerdo al análisis del documento de incidencias y la identificación del problema, se detallan dos alternativas:

4.1 Opción 1: Incrementar la potencia del hardware para que de esta manera pueda abastecer la gran demanda de los procesos enviados a ejecutar.

4.2 Opción 2: Cambiar la infraestructura actual (centralizada) por la infraestructura SAS Grid (distribuida), la cual permitirá disminuir los tiempos de ejecución de los procesos SAS.

**Tabla:** Ventajas y Desventajas de incrementar la potencia del hardware para que pueda abastecer la gran demanda de los procesos SAS enviados a ejecutar

<b>Opción 1: Incrementar la potencia del hardware para que pueda abastecer la gran demanda de los procesos SAS enviados a ejecutar</b>	
<b>Ventajas:</b>	<b>Desventajas:</b>
<ul style="list-style-type: none"> <li>Se ejecutarán mayor cantidad de procesos SAS.</li> </ul>	<ul style="list-style-type: none"> <li>No hay algoritmo de priorización de ejecución de los procesos.</li> <li>No existe servidor de respaldo.</li> <li>Servidor centralizado</li> </ul>

Fuente: Propia

**Tabla:** Cambiar la infraestructura centralizada por la infraestructura SAS Grid distribuida, la cual permitirá disminuir los tiempos de ejecución de los procesos SAS

<b>Opción 2: Cambiar la infraestructura actual (centralizada) por la infraestructura SAS Grid (distribuida), la cual permitirá disminuir los tiempos de ejecución de los procesos SAS</b>	
<b>Ventajas:</b>	<b>Desventajas:</b>
<ul style="list-style-type: none"> <li>• Alta Disponibilidad de la información.</li> <li>• Balanceo de carga de trabajo.</li> <li>• Disminución de los tiempos de ejecución de los procesos SAS.</li> <li>• Servidor de respaldo</li> <li>• Es flexible y escalable.</li> <li>• Algoritmo de priorización de procesos SAS</li> </ul>	<ul style="list-style-type: none"> <li>• Incrementa el uso de Jobs.</li> </ul>

Fuente: Propia

Por lo tanto, es factible como objetivo a esta problemática la opción 2 porque ofrece mejores beneficios que apoyan a los objetivos estratégicos del área de soluciones de información, los cuales se detallan a continuación:

**Tabla:** Cuadro Comparativo entre la Opción 1 y Opción 2 propuestas

<b>OPCIÓN 1</b>	<b>OPCIÓN 2</b>
- Poca Disponibilidad: Si bien se incrementa la potencia de hardware seguirá siendo un servidor centralizado latente a caídas.	- Alta Disponibilidad: La solución SAS GRID por ser un sistema distribuido ofrece alta disponibilidad.
- Carga de procesamiento Centralizada: El servidor actual esta bajo la plataforma Windows lo cual no permite el balanceo de carga.	- Balanceo de carga: El Servidor Grid Manager será el encargado de distribuir la carga entre la cantidad de nodos que tenga en su infraestructura y bajo la plataforma AIX.

<ul style="list-style-type: none"> <li>- Servidor Centralizado: Es dependiente de un único Servidor el cual ejecuta los procesos SAS y si llegase a caer se paralizarían la ejecución de los procesos SAS</li> </ul>	<ul style="list-style-type: none"> <li>- Servidor Distribuido: A diferencia de SAS que es un servidor centralizado, SAS Grid no depende de un solo servidor, sino tiene servidor de respaldo, y servidores encargados a la ejecución de los procesos SAS que permiten la compartición de recursos computacionales.</li> </ul>
	<ul style="list-style-type: none"> <li>- Mejora en los tiempos de ejecución de procesos SAS: Con el balanceo de carga y la alta disponibilidad se logrará minimizar los tiempos de ejecución de los procesos SAS.</li> </ul>
	<ul style="list-style-type: none"> <li>- Flexibilidad y Escalabilidad: Pueden añadirse servidores en la infraestructura, a medida que la carga de trabajo se incremente, sin afectar al resto de los servidores.</li> </ul>
	<ul style="list-style-type: none"> <li>- Algoritmo de priorización de ejecución de procesos SAS.</li> </ul>

Fuente: Propia

## V. Objetivo

Diseñar la infraestructura SAS Grid que optimice el tiempo de ejecución de los procesos SAS.



### ANEXO 3

## CONSIDERACIONES PARA LA MIGRACIÓN DEL PROCESOSAS PSI\_RESUMEN\_SALDO DEL ÁREA DE SOLUCIONES DE INFORMACIÓN

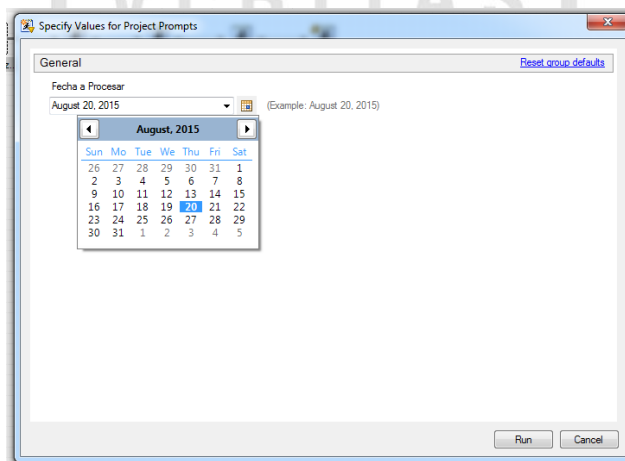
#### Estructura del proceso migrado:

Se reutilizará el proyecto guide actual, bajo el nombre  
Indice\_Estabilidad\_EM\_Final.egp.

Se mantendrá el flujo actual del proceso, para no ver afectada la rutina  
actual de ejecución.

Los parámetros del proceso se solicitarán en cada ejecución, esto evitará  
que se deba modificar manualmente el código de los programas en cada  
ejecución.

Pantalla de ejemplo de solicitud de parámetros:



#### Flujo SAS Enterprise Guide correspondiente:



#### Consideraciones para la migración del proceso:

A continuación se mencionan con el máximo nivel de detalle posible las  
consideraciones y/o modificaciones a realizar sobre las fuentes del proceso:

Precondición:

Verificar que se encuentre creada la estructura de directorios necesaria para  
el proceso.

Modificaciones sobre el proyecto Guide (Indice\_Estabilidad\_EM\_Final.egp)

Crear el parámetro FechaProc, el mismo deberá ser del tipo Date.

Todas las macros encontradas en los programas “AsignaLibs”, “Proc Mes OBS”, “OtrasMacrosGrles”, “CreaFormats”, “CalculaFREQs”, “AjustaRangos” serán registradas dentro de un nuevo programa MACROS.SAS.

### **Modificar los programas:**

#### Paso 0 – Seteo de librerías y MesProceso

Este programa incluirá el seteo de las librerías y FechaProc

Los directorios utilizados en el script y deberán ser editados cambiando “\” por “/”.

libname inf

'A:\mmgr\00 Laboratorio\02 Proyectos\201407 MatrizModelamiento\02 Resumen Saldo\Procesados';

libname data

'Z:\mmgr validacion\00 Laboratorio\02 Proyectos\201407 MatrizModelamiento\02 Resumen Saldo\Controles\test2';

libname rslt

'Z:\mmgr validacion\00 Laboratorio\02 Proyectos\201407 MatrizModelamiento\02 Resumen Saldo\Controles\test2\rslt';

DATAFILE=

"\pfilep06\mmgr\Laboratorio\Actividades\Proyectos\201506 Automatizacion ControlesMatrizVariables\01 Resumen Saldo\Controles\tabyvar.xlsx"

Asignar al programa el parámetro creado.

Inicio:

Este programa incluirá el contenido del programa “Inicio”.

Se adiciona lo siguiente para convocar las macros necesarias para el programa.

%include

“sasdata01/Soluciones\_Informacion/Resumen\_Saldoscripts/MACROS.SAS”;

Asignar al programa los parámetros creados en el paso anterior.

Eliminar del programa la declaración de la macro variable correspondiente al parámetro.

%let CODMES= '15Jul2015'd; /\*"&FechaProc"d;\*/

### Proc Mes REF:

Este programa tendrá el contenido de los programas “Proc Mes REF”.

Asignar al programa los parámetros creados.

Se adiciona lo siguiente para convocar las macros necesarias para el programa.

```
%include
```

```
“sasdata01/Soluciones_Informacion/Resumen_Saldoscripts/MACROS.SAS”;
```

### Proc Mes 1:

Este programa tendrá el contenido de los programas “ProcOBS1”,

“ProcOBS2”, “ProcOBS3”.

Asignar al programa los parámetros creados.

Se adiciona lo siguiente para convocar las macros necesarias para el programa.

```
%include
```

```
“sasdata01/Soluciones_Informacion/Resumen_Saldoscripts/MACROS.SAS”;
```

### Junta 3 Meses:

Este programa tendrá el contenido de los programas “JuntaCompares”.

Asignar al programa los parámetros creados.

Se adiciona lo siguiente para convocar las macros necesarias para el programa.

```
%include
```

```
“sasdata01/Soluciones_Informacion/Resumen_Saldoscripts/MACROS.SAS”;
```

### **Opciones Grid:**

A continuación se mencionan las modificaciones a realizar sobre los programas con el fin de utilizar funciones Grid.

Líneas Generales:

Todos los programas que componen el proceso, deberán tener añadida las siguientes líneas al inicio:

### **Activación servicio Grid:**

```
%let rc=%sysfunc(grdsvc_enable(_all_,server=SASApp));
```

```
options autosignon;
```

Variables adicionales para ejecución Jobs Grid:

```
options emailhost=("172.29.64.20" port=25);
%let mailaddr= S40073@CREDITO.BCP.COM.PE;
```

### **Definición de Jobs Grid:**

Se detallan los Jobs Grid identificados para el proceso.

#### Inicio

En el programa, se creará un nuevo Job, que incluirá todas las transformaciones realizadas en el programa, dado que la forma en que está desarrollado el programa, no es posible paralelizar tareas.

Job\_asg:

Comienzo del Jobs:

Al inicio del programa, justo antes del Proc Import que realiza la importación del excel, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/
%syslput LotRecs =&LotRecs. /remote=job_asg;
%syslput FilePref=&FilePref /remote= job_asg;
%syslput Tags=&Tags. /remote= job_asg;
%syslput TextoProceso=&TextoProceso. /remote= job_asg;
%syslput MESR=&MESR. /remote= job_asg;
%syslput CODMES=&CODMES.. /remote= job_asg;

/*Inicializar fecha de comienzo del job*/
data _null_;
tempdate=put(date(),ddmmyyn8.);
temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;
/*Definición del Job*/
rsubmit job_asg wait=no cmacvar=job_asg sysrputsync=yes inheritlib=(data)
log="/sasdata01/Soluciones_Informacion/Indice_Estabilidad/logs/Ejecucion_j
ob_ASG_&currdate..log" new
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio
Ejecucion Job_ASG ";
```

```
options fullstimer mprint mlogic;
```

```
...
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_asg;
```

### Proc Mes REF:

En el programa, se creará un nuevo Job, dado que la forma en que está desarrollado el programa, no es posible paralelizar tareas.

Job\_SQL1:

Comienzo del Jobs:

Al inicio del programa, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/  
%syslput FilePref=&FilePref /remote=job_SQL1;  
%syslput MaxLoteC =&MaxLoteC. /remote=job_SQL1;  
%syslput MaxLoteD=&MaxLoteD. /remote=job_SQL1;  
%syslput Tags=&Tags. /remote=job_SQL1;  
%syslput TextoProceso=&TextoProceso. /remote=job_SQL1;  
%syslput MESR=&MESR. /remote=job_SQL1;
```

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;  
tempdate=put(date(),ddmmyyn8.);  
temptime=put(time(),b8601tm.);  
call symput('currdate',catx('_',tempdate,temptime));  
run;
```

/\*Definición del Job\*/

```
rsubmit job_comp wait=no cmacvar=job_comp sysrputsync=yes
```

```
inheritlib=(inf data)
```

```
log="/sasdata01/Soluciones_Informacion/Indice_Estabilidad/logs/Ejecucion_j
```

```
ob_sql1_&currdate..log" new
```

```
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio
```

```
Ejecucion Compares Job_sql1 ";
```

```
options fullstimer mprint mlogic;
```

...

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_sql1;
```

Proc Mes 1:

En el programa, se creará un nuevo Job, dado que la forma en que está desarrollado el programa, no es posible paralelizar tareas.

Job\_MES1:

Comienzo del Job:

Al inicio del programa, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/
```

```
%syslput FilePref=&FilePref /remote=job_MES1;
```

```
%syslput MaxLoteC =&MaxLoteC. /remote=job_MES1;
```

```
%syslput MaxLoteD=&MaxLoteD. /remote=job_MES1;
```

```
%syslput Tags=&Tags. /remote=job_MES1;
```

```
%syslput TextoProceso=&TextoProceso. /remote=job_MES1;
```

```
%syslput MESR=&MESR. /remote=job_MES1;
```

```

/*Inicializar fecha de comienzo del job*/
data _null_;
tempdate=put(date(),ddmmyyn8.);
temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;

```

```

/*Definición del Job*/

```

```

rsubmit job_comp wait=no cmacvar=job_comp sysrputsync=yes
inheritlib=(inf data)
log="/sasdata01/Soluciones_Informacion/Indice_Estabilidad/logs/Ejecucion_j
ob_mes1_&currdate..log" new
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio
Ejecucion job_Mes1 ";
options fullstimer mprint mlogic;

```

...

```

/*Definición del Job*/

```

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```

endrsubmit;

```

```

Job_MES2:

```

1. Comienzo del Job:

Al inicio del programa, añadir las siguientes líneas:

```

/*Incluir variables que se utilizan dentro del job*/
%syslput FilePref=&FilePref /remote=job_MES2;
%syslput MaxLoteC =&MaxLoteC. /remote=job_MES2;
%syslput MaxLoteD=&MaxLoteD. /remote=job_MES2;
%syslput Tags=&Tags. /remote=job_MES2;
%syslput TextoProceso=&TextoProceso. /remote=job_MES2;
%syslput MESR=&MESR. /remote=job_MES2;

```

```

/*Inicializar fecha de comienzo del job*/

```

```

data _null_;
tempdate=put(date(),ddmmyyn8.);

```

```

temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;

```

*/\*Definición del Job\*/*

```

rsubmit job_comp wait=no cmacvar=job_comp sysrputsync=yes
inheritlib=(inf data)
log="/sasdata01/Soluciones_Informacion/Indice_Estabilidad/logs/Ejecucion_j
ob_mes2_&currdate..log" new
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio
Ejecucion job_Mes2 ";
options fullstimer mprint mlogic;

```

...

2. Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrssubmit;
```

Job\_MES3:

1. Comienzo del Job:

Al inicio del programa, añadir las siguientes líneas:

```

/*Incluir variables que se utilizan dentro del job*/
%syslput FilePref=&FilePref /remote=job_MES3;
%syslput MaxLoteC =&MaxLoteC. /remote=job_MES3;
%syslput MaxLoteD=&MaxLoteD. /remote=job_MES3;
%syslput Tags=&Tags. /remote=job_MES3;
%syslput TextoProceso=&TextoProceso. /remote=job_MES3;
%syslput MESR=&MESR. /remote=job_MES3;

```

*/\*Inicializar fecha de comienzo del job\*/*

```

data _null_;
tempdate=put(date(),ddmmyyn8.);
temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;

```



/\*Definición del Job\*/

```
rsubmit job_comp wait=no cmacvar=job_comp sysrputsync=yes
```

```
inheritlib=(inf data)
```

```
log="/sasdata01/Soluciones_Informacion/Indice_Estabilidad/logs/Ejecucion_j
```

```
ob_mes3_&currdate..log" new
```

```
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio
```

```
Ejecucion job_Mes3 ";
```

```
options fullstimer mprint mlogic;
```

...

2. Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

3. Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_MES1 job_MES2 job_MES3;
```

b. Junta 3 Meses:

En el programa, se creará un nuevo Job, dado que la forma en que está desarrollado el programa, no es posible paralelizar tareas.

Job\_SQL2:

1. Comienzo del Jobs:

Al inicio del programa, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/
```

```
%syslput FilePref=&FilePref /remote=job_SQL2;
```

```
%syslput MaxLoteC =&MaxLoteC. /remote=job_SQL2;
```

```
%syslput MaxLoteD=&MaxLoteD. /remote=job_SQL2;
```

```
%syslput Tags=&Tags. /remote=job_SQL2;
```

```
%syslput TextoProceso=&TextoProceso. /remote=job_SQL2;
```

```
%syslput MESR=&MESR. /remote=job_SQL2;
```

```

/*Inicializar fecha de comienzo del job*/
data _null_;
tempdate=put(date(),ddmmyyn8.);
temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;

```

```

/*Definición del Job*/

```

```

rsubmit job_comp wait=no cmacvar=job_comp sysrputsync=yes
inheritlib=(inf data)
log="/sasdata01/Soluciones_Informacion/Indice_Estabilidad/logs/Ejecucion_j
ob_sql2_&currdate..log" new
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio
Ejecucion Compares Job_sql2 ";

```

```

options fullstimer mprint mlogic;

```

...

2. Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```

endrsubmit;

```

3. Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```

waitfor _all_ job_sql2;

```

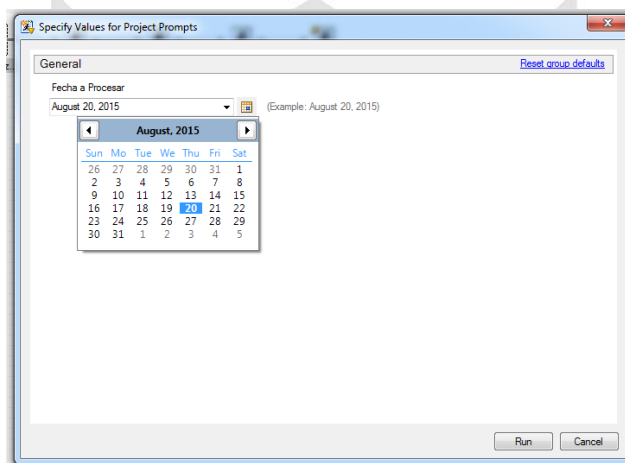
## ANEXO 4

### CONSIDERACIONES PARA LA MIGRACIÓN DEL PROCESOS SAS COMPORTAMIENTO\_Y\_ALERTAS DEL ÁREA DE SOLUCIONES DE INFORMACIÓN

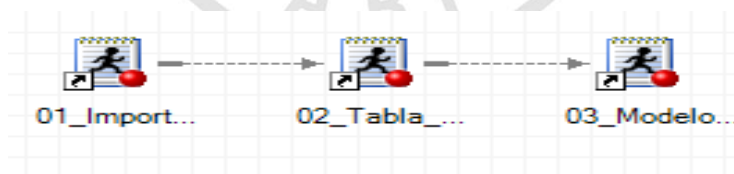
#### Estructura del proceso migrado:

Se generará el proyecto guide BCP3\_Comportamientos\_y\_ Alertas.egp.  
Se generará un flujo encadenando los script en el orden de ejecución actual.  
Se añadirá un parámetro que indique el mes y año correspondiente al proceso, esto evitará que se deba modificar manualmente el código de los programas en cada ejecución.

Pantalla de ejemplo de solicitud de parámetros:



#### Flujo SAS Enterprise Guide correspondiente:



#### Consideraciones para la migración del proceso:

A continuación se mencionan con el máximo nivel de detalle posible las consideraciones y/o modificaciones a realizar sobre las fuentes del proceso:

Precondición:

Verificar que se encuentre creada la estructura de directorios necesaria para el proceso.

Modificaciones sobre el proyecto Guide (BCP3\_Comportamientos\_y\_Alertas.egp)

Crear el parámetro aniomes, el mismo deberá ser del tipo Date.

Agregar el cálculo del mes de proceso.

```
%LET ANIOMES = %SYSFUNC(PUTN(&aniomes, YMMN6));
```

Modificar los programas:

01 Importacion Bases Guide.sas:

Los directorios utilizados en el script y deberán ser editados cambiando “\” por “/”.

```
libname prcs
```

```
"\\psasp04\SI ADR PROD\03 Procesos\Scores\06 Rating BCP3 CyA\201510\INPUTS";
```

Asignar al programa el parámetro creado.

02 Tabla Unica Guide.sas:

Asignar al programa el parámetro creado.

Eliminar las líneas correspondientes al seteo de librerías.

```
libname input
```

```
"\\psasp04\SI ADR PROD\03 Procesos\Scores\06 Rating BCP3 CyA\201510\inputs";
```

```
libname proceso
```

```
"\\psasp04\SI ADR PROD\03 Procesos\Scores\06 Rating BCP3 CyA\201510\proceso";
```

```
libname output
```

```
"\\psasp04\SI ADR PROD\03 Procesos\Scores\06 Rating BCP3 CyA\201510";
```

03 Modelo guide.sas:

Asignar al programa el parámetro creado.

Eliminar las líneas correspondientes al seteo de librerías.

```
libname input
```

```
"\\psasp04\SI ADR PROD\03 Procesos\Scores\06 Rating BCP3 CyA\201510\inputs";
```

```
libname proceso
```

```
"\\psasp04\SI ADR PROD\03 Procesos\Scores\06 Rating BCP3 CyA\201510\proceso";
```

libname output

"\\psasp04\SI ADR PROD\03 Procesos\Scores\06 Rating BCP3 CyA\201510";

Modificar los nombres de las tablas generadas, utilizando el parámetro ANIOMES.

~~DATA output.alertas\_comp\_scores\_201510~~

DATA output.alertas\_comp\_scores\_&CODMES

~~data output.comp\_variables\_201510~~

data output.comp\_variables\_&CODMES

~~data output.alertas\_variables\_201510~~

data output.alertas\_variables\_&CODMES

### **Opciones Grid:**

A continuación se mencionan las modificaciones a realizar sobre los programas con el fin de utilizar funciones Grid.

Líneas Generales:

Todos los programas que componen el proceso, deberán tener añadida las siguientes líneas al inicio:

Activación servicio Grid:

```
%let rc=%sysfunc(grdsvc_enable(_all_,server=SASApp));
```

```
options autosignon;
```

Variables adicionales para ejecución Jobs Grid:

```
options emailhost=("172.29.64.20" port=25);
```

```
%let mailaddr= S35105@CREDITO.BCP.COM.PE;
```

### **Definición de Jobs Grid:**

Se detallan los Jobs Grid identificados para el proceso.

01 Importacion Bases Guide.sas:

En el programa, se creará un nuevo Job, que incluirá todas las transformaciones realizadas en el programa.

Job\_Importa:

Comienzo del Jobs:

Al inicio del programa, justo antes del Proc sql inicial, añadir las siguientes líneas:

```

/*Inicializar fecha de comienzo del job*/
data _null_;
tempdate=put(date(),ddmmyyn8.);
temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;

```

/\*Definición del Job\*/

```

rsubmit job_imp wait=no cmacvar=job_imp sysrputsync=yes
inheritlib=(input)
log="/sasdata01/SI_ADR_RECU_04/Comp_Y_Alertas/logs/Importacion_job_
&currdate..log" new
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio
Ejecucion Job_Importacion ";

```

```
options fullstimer mprint mlogic;
```

...

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_imp;
```

### 02 Tabla Unica Guide:

En el programa, se creará un nuevo Job, dado que la forma en que está desarrollado el programa, no es posible paralelizar tareas.

Job\_consolida:

Comienzo del Jobs:

Al inicio del programa, añadir las siguientes líneas:

```

/*Inicializar fecha de comienzo del job*/
data _null_;
tempdate=put(date(),ddmmyyn8.);
temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;

```

/\*Definición del Job\*/

```

rsubmit job_cons wait=no cmacvar=job_cons sysrputsync=yes
inheritlib=(proceso output)
log="/sasdata01/SI_ADR_RECU_04/Comp_Y_Alertas/logs/tablaUnica_job_&
currdate..log" new
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio
Ejecucion Tabla Unica job_Cons ";
options fullstimer mprint mlogic;

```

...

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

**Ejecución del job:**

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_cons;
```

03 Modelo guide:

En el programa, se creará un nuevo Job, dado que la forma en que está desarrollado el programa, no es posible paralelizar tareas.

Job\_modelo:

Comienzo del Jobs:

Al inicio del programa, añadir las siguientes líneas:

```

/*Incluir variables que se utilizan dentro del job*/
%syslput ANIOMES =&ANIOMES.. /remote=job_mod;

```

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

```
/*Definición del Job*/
```

```
rsubmit job_mod wait=no cmacvar=job_mod sysrputsync=yes
```

```
inheritlib=(proceso output)
```

```
log="/sasdata01/SI_ADR_RECU_04/Comp_Y_Alertas/logs/Modelo_job_&cur  
rdate..log" new
```

```
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio  
Ejecucion Tabla Unica job_mod";
```

```
options fullstimer mprint mlogic;
```

```
...
```

```
Fin del job:
```

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

```
Ejecución del job:
```

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_mod;
```



## ANEXO 5

### CONSIDERACIONES PARA LA MIGRACIÓN DEL PROCESOS SAS BASE\_ESPEJO\_BURO DEL ÁREA DE SOLUCIONES DE INFORMACIÓN

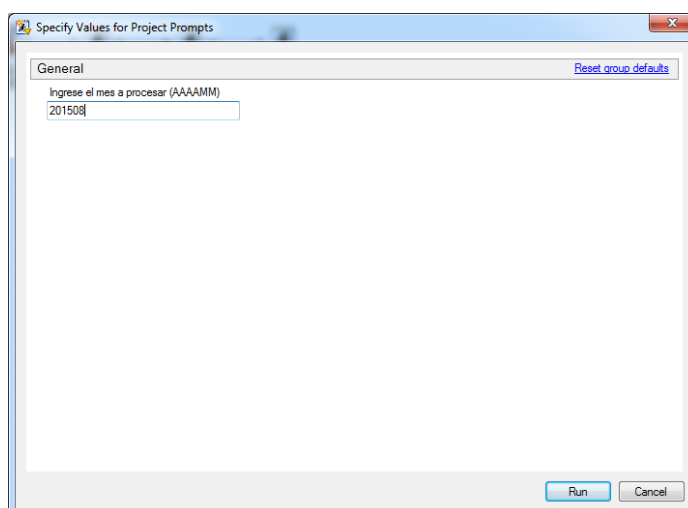
#### Estructura del proceso migrado:

El proceso será migrado a un nuevo Proyecto Guide que será generado bajo el nombre Base\_Espejo\_Buro.egp

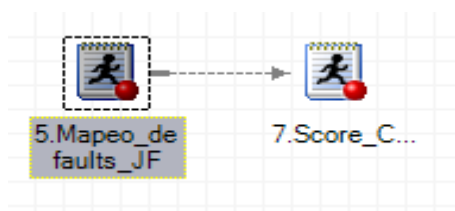
Dentro de este proyecto se generará un nuevo flujo que tendrá la secuencia actual de los programas.

Al iniciar el programa, solicitará 1 parámetro: ANIOMES que será utilizado a lo largo del proceso, sin la necesidad de modificar manualmente los programas en cada ejecución.

Pantalla de ejemplo de solicitud de parámetros:



#### Flujo SAS Enterprise Guide correspondiente:



#### Consideraciones para la migración del proceso:

A continuación se mencionan con el máximo nivel de detalle posible las consideraciones y/o modificaciones a realizar sobre las fuentes del proceso:

Precondición:

Verificar que se encuentren configuradas las librerías Oracle, que son utilizadas como Input del proceso.

Verificar que se encuentre creada la estructura de directorios necesaria para el proceso.

Modificaciones sobre el proyecto Guide (Base\_Espejo\_Buro.egp)

Crear el parámetro ANIOMES, el mismo deberá ser del tipo Date.

### **Modificar los programas:**

#### 5. Mapeo\_defaults JF:

Asignar al programa el parámetro creado.

Eliminar líneas correspondientes a conexiones al servidor, dado que en la migración, todos los procesos se ejecutarán en el servidor.

```
OPTIONS AWSDEF=0 0 100 100 MISSING=' ';
%LET IP=10.80.193.25;
%LET NSERV=PSASP04;
SIGNOFF WNT;
%LET WNT=&IP;
OPTIONS COMAMID = TCP REMOTE = WNT;
SIGNON WNT;
...
RSubmit;
ENDRSubmit;
...
PROC PRINTTO LOG="&DDIR.MAPEO_DEFAULTS_201510.LOG" NEW;
RUN; QUIT;
...
/*WNT*/ LIBNAME BCP          SLIBREF=BCP          SERVER=WNT;
/*WNT*/ LIBNAME TRABAJO     SLIBREF=WORK        SERVER=WNT;

/* CERRANDO EL LOG-FILE */

DATA _NULL_;
DD="%SYSFUNC(putn(%sysfunc(datetime()),datetime22.))";
```

```

DD=COMPRESS(DD," :");
DD=SUBSTR(DD,1,9)||"_"||SUBSTR(DD,10);
CALL SYMPUT("DDFIN",COMPRESS(DD));
RUN;
%MACRO NOECO;
%PUT *****;
%PUT INICIO DEL PROCESO: &DDINI.;
%PUT FIN DEL PROCESO: &DDFIN.;
%PUT *****;
%MEND; %NOECO;

```

Modificar los directorios utilizados en el script.

```

%LET
DDIR=\\psasp04\SI_ADR_PROD\03_Procesos\Scores\07_Credit_Score_Buro\Temp_Buro;
LIBNAME BCPINC
"\\PSASP04\SI_ADR_MV_02\00.LABORATORIO\201407_MATRIZMODELAMIENTO\01_MAPEODEFAULTS\INC";

```

En todos los directorios “relativos” se debe cambiar “\” por “/”.

### 7 Score Calibrado

Asignar al programa el parámetro creado.

Modificar las tablas que utilizan el parámetro ANIOMES

```

SET DB_MMGR.TU_BURO_EN_PROC_201509;
SET DB_MMGR.TU_BURO_EN_PROC_&CODMES;

```

Eliminar líneas correspondientes a conexiones al servidor, dado que en la migración, todos los procesos se ejecutarán en el servidor.

```

OPTIONS AWSDEF=0 0 100 100 MISSING=' ';
%LET IP=10.80.193.25;
%LET NSERV=PSASP04;
SIGNOFF WNT;
%LET WNT=&IP 23;

```

```

OPTIONS COMAMID = TCP REMOTE = WNT;
SIGNON WNT;
RSUBMIT;
ENDRSUBMIT;

/*WNT*/ LIBNAME BCP          SLIBREF=BCP          SERVER=WNT;
/*WNT*/ LIBNAME DB_MMGR    SLIBREF=DB_MMGR    SERVER=WNT;

```

```

/* CERRANDO EL LOG-FILE */
DATA _NULL_;
DD="%SYSFUNC(PUTN(%SYSFUNC(DATETIME()),DATETIME22.))";
DD=COMPRESS(DD," :");
DD=SUBSTR(DD,1,9)||"_"||SUBSTR(DD,10);
CALL SYMPUT("DDFIN",COMPRESS(DD));
RUN;
%MACRO NOECO;
%PUT *****;
%PUT INICIO DEL PROCESO: &DDINI.;
%PUT FIN DEL PROCESO: &DDFIN.;
%PUT *****;
%MEND; %NOECO;
PROC PRINTTO; RUN;
%NOECO;

```

Modificar los directorios utilizados en el script.

```

LIBNAME BCP
"\\PSASP04\MMGR5\00.LABORATORIO\201407_MATRIZMODELAMIENTO
\01_MAPEODEFAULTS\PROCESAMIENTO\201509";

```

En todos los directorios "relativos" se debe cambiar "\" por "/".

### Opciones Grid:

A continuación se mencionan las modificaciones a realizar sobre los programas con el fin de utilizar funciones Grid.

Líneas Generales:

Todos los programas que componen el proceso, deberán tener añadida las siguientes líneas al inicio:

Activación servicio Grid:

%let rc=%sysfunc(grdsvc\_enable( all ,server=SASApp));

options autosignon;

Variables adicionales para ejecución Jobs Grid:

options emailhost=("172.29.64.20" port=25);

%let mailaddr= S40548@CREDITO.BCP.COM.PE;

### **Definición de Jobs Grid:**

Se detallan los Jobs Grid identificados para el proceso.

5. Mapeo\_defaults\_JF:

En el programa, se creará 1 job grid, que incluirá tanto las descargas de Oracle como las posteriores cargas.

Job\_dJF:

Comienzo del Jobs:

Previo a realizar la primera descarga de Oracle y posterior al seteo de las librerías:

*/\*Incluir variables que se utilizan dentro del job\*/*

*%syslput ANIOMES=&ANIOMES. /remote=job\_dJF;*

*%syslput ANIOMES1=&ANIOMES1. /remote=job\_dJF;*

*/\*Inicializar fecha de comienzo del job\*/*

*data \_null\_;*

*tempdate=put(date(),ddmmyyn8.);*

*temptime=put(time(),b8601tm.);*

*call symput('currdate',catx('\_',tempdate,temptime));*

*run;*

*/\*Definición del Job\*/*

*rsubmit job\_djf wait=no cmacvar=job\_djf sysrputsync=yes inheritlib=(BCP BCPINC BCPINC1)*

*log="/sasdata01/SI\_ADR\_RECU\_04/Base\_Espejo\_Buro/Soluciones\_Información\_job\_djf\_&currdate..log " new*

```
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio
Ejecucion Sol. Inf. Buro Job_djf ";
options fullstimer mprint mlogic;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_djf;
```

7 Score Calibrado:

En el programa, se creará 1 job grid, que incluirá tanto las descargas de Oracle como las posteriores cargas.

Job\_scr:

Comienzo del Jobs:

Previo a realizar la primera descarga de Oracle y posterior al seteo de las librerías:

```
/*Incluir variables que se utilizan dentro del job*/
```

```
%syslput ANIOMES=&ANIOMES. /remote=job_scr;
```

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

```
/*Definición del Job*/
```

```
rsubmit job_scr wait=no cmacvar=job_scr sysrputsync=yes inheritlib=(BCP)
```

```
log="/sasdata01/SI_ADR_RECU_04/Base_Espejo_Buro/logs/soluciones_info
rmación_job_scr_&CURRDATE..log " new
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio
Ejecucion Sol. Inf. Buro Job_scr ";
;
options fullstimer mprint mlogic;
```

Fin del job:

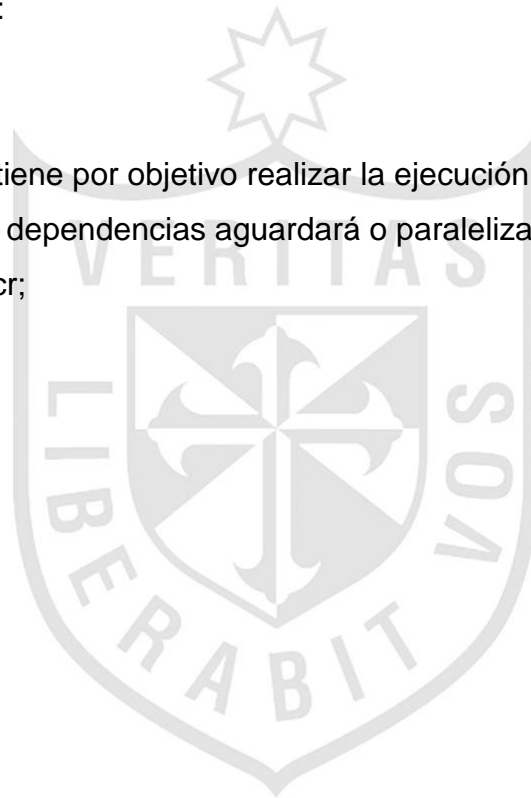
La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_job_scr;
```



**ANEXO 6**  
**CONSIDERACIONES PARA LA MIGRACIÓN DEL PROCESOS SAS**  
**ESTADOS\_FINANCIEROS DEL ÁREA DE SOLUCIONES DE**  
**INFORMACIÓN**

**Estructura del proceso migrado:**

Se generará un proyecto Guide con 3 flujos diferentes, uno para cada subproceso.

Cada uno de los flujos tendrá el programa SAS correspondiente.

**Consideraciones para la migración del proceso:**

A continuación se mencionan con el máximo nivel de detalle posible las consideraciones y/o modificaciones a realizar sobre las fuentes del proceso:

Precondición:

Verificar que se encuentre configuradas las librerías Oracle, que son utilizadas como Input del proceso.

Verificar que se encuentre creada la estructura de directorios necesaria para el proceso.

Modificaciones sobre el proyecto Guide

**Modificar los programas:**

02\_Calculo\_Modelo\_bcp3\_eeff (Inc Analizados):

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

libname bcp3

"\\psasp04\SI ADR PROD\03 Procesos\Scores\05 Rating BCP3 EEFF\proceso\_construccion\201511";

02\_Calculo\_Modelo\_bcp3\_eeff (Construcción):

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

libname bcp3

"\\psasp04\SI ADR PROD\03 Procesos\Scores\05 Rating BCP3 EEFF\proceso\_construccion\201511";

02\_Calculo\_Modelo\_bcp3\_eeff (Total):

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.



libname bcp3

"\\psasp04\SI ADR PROD\03 Procesos\Scores\05 Rating BCP3 EEFF\proceso\_inc\_analizados\201512":

### **Opciones Grid:**

A continuación se mencionan las modificaciones a realizar sobre los programas con el fin de utilizar funciones Grid.

Líneas Generales:

Todos los programas que componen el proceso, deberán tener añadida las siguientes líneas al inicio:

Activación servicio Grid:

```
%let rc=%sysfunc(grdsvc_enable(_all_,server=SASApp));  
options autosignon;
```

Variables adicionales para ejecución Jobs Grid:

```
options emailhost=("172.29.64.20" port=25);  
%let mailaddr= S35105@CREDITO.BCP.COM.PE;
```

### **Definición de Jobs Grid:**

Se detallan los Jobs Grid identificados para el proceso.

02\_Calculo\_Modelo\_bcp3\_eeff (Inc\_Analizados):

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_Inc:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/  
data _null_;  
tempdate=put(date(),ddmmyyn8.);  
temptime=put(time(),b8601tm.);  
call symput('currdate',catx('_',tempdate,temptime));  
run;  
/*Definición del Job*/  
rsubmit job_Inc wait=no cmacvar=job_Extr sysrputsync=yes
```

```
log="/sasdata01/SI_ADR_RECU_04/EEFF/logs/Inc_Analizados_job_&currdate..log " new
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio
Ejecucion Inc_analizados Job_Inc ";
;
options fullstimer mprint mlogic;
```

---

~~Proc sql~~

----

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_Inc;
```

02\_Calculo\_Modelo\_bcp3\_eeff (Construcción):

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_Cons:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

```
/*Definición del Job*/
```

```
rsubmit job_Cons wait=no cmacvar=job_Cons sysrputsync=yes
```

```

log="/sasdata01/SI_ADR_RECU_04/EEFF/logs/Construccion_job_&currdate.
.log " new
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio
Ejecucion Construccion Job_cons ";
;
options fullstimer mprint mlogic;
...
Proc sql
....

```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_cons;
```

02\_Calculo\_Modelo\_bcp3\_eeff (Total):

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_tot:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

```
/*Definición del Job*/
```

```
rsubmit job_Tot wait=no cmacvar=job_tot sysrputsync=yes
```

```
log="/sasdata01/SI_ADR_RECU_04/EEFF/logs/Total_job_&currdate..log "  
new  
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio  
Ejecucion Total Job_Tot ";  
;  
options fullstimer mprint mlogic;  
...  
Proc sql  
....
```

Fin del job:

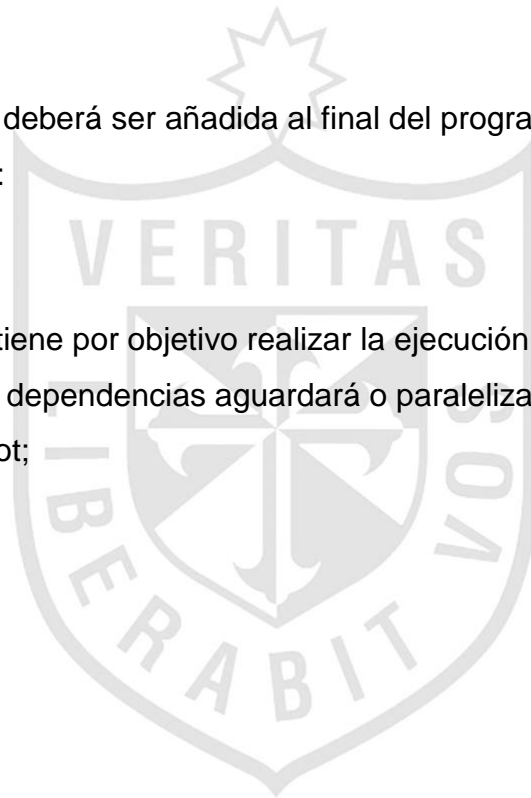
La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsbmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_Tot;
```



## ANEXO 7

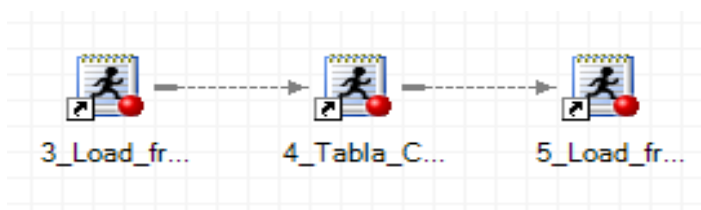
### CONSIDERACIONES PARA LA MIGRACIÓN DEL PROCESOS SAS APPLICANT\_CCO DEL ÁREA DE SOLUCIONES DE INFORMACIÓN

#### Estructura del proceso migrado:

Se generará el proyecto guide Applicant\_CCO.egp.

Se generará un flujo encadenando los script en el orden de ejecución actual.

Flujo SAS Enterprise Guide correspondiente:



#### Consideraciones para la migración del proceso:

A continuación se mencionan con el máximo nivel de detalle posible las consideraciones y/o modificaciones a realizar sobre las fuentes del proceso:

Precondición:

Verificar que se encuentre configuradas las librerías Oracle, que son utilizadas como Input del proceso.

Verificar que se encuentre creada la estructura de directorios necesaria para el proceso.

Modificaciones sobre el proyecto Guide

#### Modificar los programas:

##### 3 Load from Oracle to SAS

Eliminar líneas correspondientes a conexiones al servidor, dado que en la migración, todos los procesos se ejecutarán en el servidor.

```
OPTIONS AWSDEF=0 0 100 100 MISSING=' ';
```

```
%LET IP=10.80.193.25;
```

```
%LET NSERV=PSASP04;
```

```
SIGNOFF WNT;
```

```
%LET WNT=&IP 23;
```

```
OPTIONS COMAMID = TCP REMOTE = WNT;
```

```
SIGNON WNT;
```

...

```

RSUBMIT;
ENDRSUBMIT;
...
LIBNAME DATA SLIBREF=DATA SERVER=WNT;
LIBNAME INPUT SLIBREF=INPUT SERVER=WNT;
RSUBMIT;
ENDRSUBMIT;

```

Modificar los directorios utilizados en el script.

En todos los directorios "relativos" se debe cambiar "\" por "/".

```

LIBNAME INPUT
"\\PSASP04\SI_ADR_PROD\03_PROCESOS\SCORES\01_FT_NUEVO_AP
PLICANT_CONSUMO\201512\INPUT";

```

#### 4 Tabla Calculadora RCC2

Modificar los directorios utilizados en el script.

En todos los directorios "relativos" se debe cambiar "\" por "/".

```

LIBNAME INPUT
"\\PSASP04\SI_ADR_PROD\03_PROCESOS\SCORES\01_FT_NUEVO_AP
PLICANT_CONSUMO\201512\INPUT";
LIBNAME INTERM
"\\PSASP04\SI_ADR_PROD\03_PROCESOS\SCORES\01_FT_NUEVO_AP
PLICANT_CONSUMO\201512\INTERM";
LIBNAME OUTPT
"\\PSASP04\SI_ADR_PROD\03_PROCESOS\SCORES\01_FT_NUEVO_AP
PLICANT_CONSUMO\201512\OUTPT";

```

#### 5 Load from SAS to Oracle

Eliminar líneas correspondientes a conexiones al servidor, dado que en la migración, todos los procesos se ejecutarán en el servidor.

```

OPTIONS AWSDEF=0 0 100 100 MISSING=' ';
%LET IP=10.80.193.25;
%LET NSERV=PSASP04;

```

```

SIGNOFF WNT;
%LET WNT=&IP 23;
OPTIONS COMAMID = TCP REMOTE = WNT;
SIGNON WNT;
...
RSUBMIT;
ENDRSUBMIT;
...
RSUBMIT;
ENDRSUBMIT;

```

Modificar los directorios utilizados en el script.

En todos los directorios "relativos" se debe cambiar "\" por "/".

```

LIBNAME BCP
"\Psasp04\SI_ADR_PROD\03_PROCESOS\SCORES\01_FT_NUEVO_APP
LICANT_CONSUMO\201512\OUTPT";

```

### **Opciones Grid:**

A continuación se mencionan las modificaciones a realizar sobre los programas con el fin de utilizar funciones Grid.

Líneas Generales:

Todos los programas que componen el proceso, deberán tener añadida las siguientes líneas al inicio:

Activación servicio Grid:

```

%let rc=%sysfunc(grdsvc_enable(_all_,server=SASApp));
options autosignon;

```

Variables adicionales para ejecución Jobs Grid:

```

options emailhost=("172.29.64.20" port=25);
%let mailaddr= s40548@CREDITO.BCP.COM.PE;

```

Definición de Jobs Grid:

Se detallan los Jobs Grid identificados para el proceso.

3\_Load\_from\_Oracle\_to\_SAS:

En el programa, se creará 1 job grid, que incluirá las descargas de Oracle.

Job\_desc:

Comienzo del Job:

Al inicio del programa, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/  
%LET ANIOMES = "&CODMES"D;  
%LET ANIOMES = %SYSFUNC(PUTN(&ANIOMES, YYMMN6.));
```

Previo a realizar la primera descarga de Oracle y posterior al seteo de las librerías:

```
/*Inicializar fecha de comienzo del job*/  
data _null_;  
tempdate=put(date(),ddmmyyn8.);  
temptime=put(time(),b8601tm.);  
call symput('currdate',catx('_',tempdate,temptime));  
run;  
/*Definición del Job*/  
rsubmit job_desc wait=no cmacvar=job_desc sysrputsync=yes inheritlib=(  
INPUT)  
log="/sasdata01/SI_ADR_RECU_04/Applicant_CCO/logs/Soluciones_Inform  
ación_job_desc_&currdate..log" new  
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio  
Ejecucion Sol. Inf. Applicat CCO Job_desc ";  
;  
options fullstimer mprint mlogic;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```



Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.  
waitfor \_all\_ job\_desc;

4\_Tabla\_Calculadora\_RCC2:

En el programa, se creará 1 job grid, que incluirá las descargas de Oracle.  
Job\_calc:

Comienzo del Jobs:

Previo a realizar la primera descarga de Oracle y posterior al seteo de las librerías:

```
/*Inicializar fecha de comienzo del job*/  
data _null_;  
tempdate=put(date(),ddmmyyn8.);  
temptime=put(time(),b8601tm.);  
call symput('currdate',catx('_',tempdate,temptime));  
run;  
  
/*Definición del Job*/  
rsubmit job_desc wait=no cmacvar=job_desc sysrputsync=yes  
inheritlib=(INTERM INPUT OUTPT)  
log="/sasdata01/SI_ADR_RECU_04/Applicant_CCO/logs/Soluciones_Información_job_carga_&currdate..log " new  
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio  
Ejecucion Sol. Inf. Applicat CCO Job_calc";  
;  
options fullstimer mprint mlogic;  
Fin del job:  
La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:  
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_ calc;
```

5\_Load\_from\_SAS\_to\_Oracle:

En el programa, se creará 1 job grid, que incluirá las descargas de Oracle.

Job\_carga:

Comienzo del Jobs:

Al inicio del programa, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/
```

```
%LET ANIOMES = "&CODMES"D;
```

```
%LET ANIOMES = %SYSFUNC(PUTN(&ANIOMES, YYMMN6.));
```

Previo a realizar la primera descarga de Oracle y posterior al seteo de las librerías:

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

```
/*Definición del Job*/
```

```
rsubmit job_carg wait=no cmacvar=job_carg sysrputsync=yes inheritlib=(  
BCP)
```

```
log="/sasdata01/SI_ADR_RECU_04/Applicant_CCO/logs/Soluciones_Inform  
ación_job_carga_&currdate..log" new
```

```
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio  
Ejecucion Sol. Inf. Applicat CCO Job_carga";
```

```
;
```

```
options fullstimer mprint mlogic;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrssubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_carg;
```



## ANEXO 8

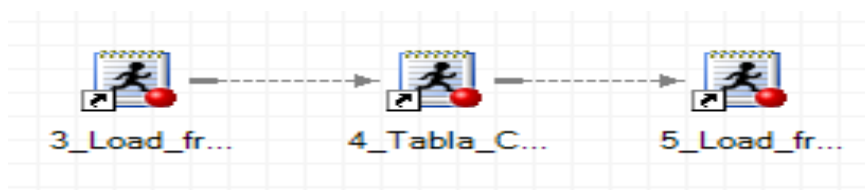
### CONSIDERACIONES PARA LA MIGRACIÓN DEL PROCESOS SAS APPLICANT\_PYME DEL ÁREA DE SOLUCIONES DE INFORMACIÓN

#### Estructura del proceso migrado:

Se generará el proyecto guide Applicant\_CCO.egp.

Se generará un flujo encadenando los script en el orden de ejecución actual.

Flujo SAS Enterprise Guide correspondiente:



#### Consideraciones para la migración del proceso:

A continuación se mencionan con el máximo nivel de detalle posible las consideraciones y/o modificaciones a realizar sobre las fuentes del proceso:

Precondición:

Verificar que se encuentre configuradas las librerías Oracle, que son utilizadas como Input del proceso.

Verificar que se encuentre creada la estructura de directorios necesaria para el proceso.

Modificaciones sobre el proyecto Guide

#### Modificar los programas:

##### 3 Load from Oracle to SAS

Eliminar líneas correspondientes a conexiones al servidor, dado que en la migración, todos los procesos se ejecutarán en el servidor.

```
OPTIONS AWSDEF=0 0 100 100 MISSING=' ';
```

```
%LET IP=10.80.193.25;
```

```
%LET NSERV=PSASP04;
```

```
SIGNOFF WNT;
```

```
%LET WNT=&IP 23;
```

```
OPTIONS COMAMID = TCP REMOTE = WNT;
```

```
SIGNON WNT;
```

...

```

RSUBMIT;
ENDRSUBMIT;
...
LIBNAME DATA SLIBREF=DATA SERVER=WNT;
LIBNAME INPUT SLIBREF=INPUT SERVER=WNT;
RSUBMIT;
ENDRSUBMIT;

```

Modificar los directorios utilizados en el script.

En todos los directorios “relativos” se debe cambiar “\” por “/”.

```

LIBNAME INPUT
"\\PSASP04\SI_ADR_PROD\03_PROCESOS\SCORES\01_FT_NUEVO_AP
PLICANT_CONSUMO\201512\INPUT";

```

#### 4 Tabla Calculadora RCC2

Modificar los directorios utilizados en el script.

En todos los directorios “relativos” se debe cambiar “\” por “/”.

```

LIBNAME INPUT
"\\PSASP04\SI_ADR_PROD\03_PROCESOS\SCORES\01_FT_NUEVO_AP
PLICANT_CONSUMO\201512\INPUT";
LIBNAME INTERM
"\\PSASP04\SI_ADR_PROD\03_PROCESOS\SCORES\01_FT_NUEVO_AP
PLICANT_CONSUMO\201512\INTERM";
LIBNAME OUTPT
"\\PSASP04\SI_ADR_PROD\03_PROCESOS\SCORES\01_FT_NUEVO_AP
PLICANT_CONSUMO\201512\OUTPT";

```

#### 5 Load from SAS to Oracle

Eliminar líneas correspondientes a conexiones al servidor, dado que en la migración, todos los procesos se ejecutarán en el servidor.

```

OPTIONS AWSDEF=0 0 100 100 MISSING=' ';
%LET IP=10.80.193.25;
%LET NSERV=PSASP04;
SIGNOFF WNT;

```

```
%LET WNT=&IP 23;
OPTIONS COMAMID = TCP REMOTE = WNT;
SIGNON WNT;
...
Rsubmit;
ENDRsubmit;
...
Rsubmit;
ENDRsubmit;
```

Modificar los directorios utilizados en el script.

En todos los directorios "relativos" se debe cambiar "\" por "/".

```
LIBNAME BCP
```

```
"\\Psasp04\SI_ADR_PROD\03_PROCESOS\SCORES\01_FT_NUEVO_APP
LICANT_CONSUMO\201512\OUTPT";
```

Opciones Grid:

A continuación se mencionan las modificaciones a realizar sobre los programas con el fin de utilizar funciones Grid.

Líneas Generales:

Todos los programas que componen el proceso, deberán tener añadida las siguientes líneas al inicio:

Activación servicio Grid:

```
%let rc=%sysfunc(grdsvc_enable(_all_,server=SASApp));
options autosignon;
```

Variables adicionales para ejecución Jobs Grid:

```
options emailhost=("172.29.64.20" port=25);
%let mailaddr= s40548@CREDITO.BCP.COM.PE;
```

Definición de Jobs Grid:

Se detallan los Jobs Grid identificados para el proceso.

3\_Load\_from\_Oracle\_to\_SAS:

En el programa, se creará 1 job grid, que incluirá las descargas de Oracle.

Job\_desc:

Comienzo del Job:

Al inicio del programa, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/  
%LET ANIOMES = "&CODMES"D;  
%LET ANIOMES = %SYSFUNC(PUTN(&ANIOMES, YYMMN6.));
```

Previo a realizar la primera descarga de Oracle y posterior al seteo de las librerías:

```
/*Inicializar fecha de comienzo del job*/  
data _null_;  
tempdate=put(date(),ddmmyyn8.);  
temptime=put(time(),b8601tm.);  
call symput('currdate',catx('_',tempdate,temptime));  
run;
```

/\*Definición del Job\*/

```
rsubmit job_desc wait=no cmacvar=job_desc sysrputsync=yes inheritlib=(  
INPUT)  
log="/sasdata01/SI_ADR_RECU_04/Applicant_CCO/logs/Soluciones_Inform  
ación_job_desc_&currdate..log" new  
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio  
Ejecucion Sol. Inf. Applicat CCO Job_desc ";  
;  
options fullstimer mprint mlogic;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.  
waitfor \_all\_ job\_desc;

4\_Tabla\_Calculadora\_RCC2:

En el programa, se creará 1 job grid, que incluirá las descargas de Oracle.  
Job\_calc:

Comienzo del Jobs:

Previo a realizar la primera descarga de Oracle y posterior al seteo de las librerías:

```
/*Inicializar fecha de comienzo del job*/  
data _null_;  
    tempdate=put(date(),ddmmyyn8.);  
    temptime=put(time(),b8601tm.);  
    call symput('currdate',catx('_',tempdate,temptime));  
run;  
  
/*Definición del Job*/  
rsubmit job_desc wait=no cmacvar=job_desc sysrputsync=yes  
inheritlib=(INTERM INPUT OUTPT)  
log="/sasdata01/SI_ADR_RECU_04/Applicant_CCO/logs/Soluciones_Inform  
ación_job_carga_&currdate..log " new  
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio  
Ejecucion Sol. Inf. Applicat CCO Job_calc";  
;  
options fullstimer mprint mlogic;  
Fin del job:  
La siguiente línea, deberá ser añadida al final del programa para identificar la  
finalización del job:  
endrsubmit;
```



Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_ calc;
```

5\_Load\_from\_SAS\_to\_Oracle:

En el programa, se creará 1 job grid, que incluirá las descargas de Oracle.

Job\_carga:

Comienzo del Jobs:

Al inicio del programa, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/
```

```
%LET ANIOMES = "&CODMES"D;
```

```
%LET ANIOMES = %SYSFUNC(PUTN(&ANIOMES, YYMMN6.));
```

Previo a realizar la primera descarga de Oracle y posterior al seteo de las librerías:

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

```
/*Definición del Job*/
```

```
rsubmit job_carg wait=no cmacvar=job_carg sysrpustsync=yes inheritlib=(  
BCP)
```

```
log="/sasdata01/SI_ADR_RECU_04/Applicant_CCO/logs/Soluciones_Inform  
ación_job_carga_&currdate..log" new
```

```
notify="&mailaddr" subject=" Notificacion desde &syshostname: Inicio  
Ejecucion Sol. Inf. Applicat CCO Job_carga";
```

```
;
```

```
options fullstimer mprint mlogic;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrssubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_carg;
```



## ANEXO 9

### CONSIDERACIONES PARA LA MIGRACIÓN DEL PROCESOS SAS BASE\_PARÁMETROS DEL ÁREA DE SOLUCIONES DE INFORMACIÓN

#### Consideraciones para la migración del proceso:

A continuación se mencionan con el máximo nivel de detalle posible las consideraciones y/o modificaciones a realizar sobre las fuentes del proceso:

Precondición:

Verificar que se encuentre configuradas las librerías Oracle, que son utilizadas como Input del proceso.

Verificar que se encuentre creada la estructura de directorios necesaria para el proceso.

#### Modificaciones sobre el proyecto Guide:

Modificar los programas:

03\_MinoristaConsumo\_PARAM\_NEW.sas:

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

LIBNAME SEG

\\psasp04\si\_adr\_PROD\03\_Procesos\Parametros\_MINORISTA\Proceso\_p  
aralelo\_201512\CCO';

03\_MinoristaHipotecario\_PARAM\_NEW\_lgd\_octubre.sas:

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

LIBNAME SEG

\\psasp04\si\_adr\_PROD\03\_Procesos\Parametros\_MINORISTA\Proceso\_p  
aralelo\_201512\HIP';

03\_MinoristaPymeNoRev\_PARAM\_NEW.sas:

Agregar parámetro Fecha, modificando el seteo de la macroVariable &MES.

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

LIBNAME SEG

\\psasp04\si\_adr\_PROD\03\_Procesos\Parametros\_MINORISTA\Proceso\_p  
aralelo\_201512\pynre';

03\_MinoristaPymeRev\_PARAM\_NEW.sas:

Agregar parámetro Fecha, modificando el seteo de la macroVariable &MES.

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

LIBNAME SEG

'\psasp04\si adr PROD\03 Procesos\Parametros MINORISTA\Proceso p  
aralelo 201512\PYREV':

03\_MinoristaTarjeta2 PARAM NEW.sas:

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

LIBNAME SEG

'\psasp04\si adr PROD\03 Procesos\Parametros MINORISTA\Proceso p  
aralelo 201512\TC':

03\_MinoristaVehicular2 PARAM NEW.sas:

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

LIBNAME SEG

'\psasp04\si adr PROD\03 Procesos\Parametros MINORISTA\Proceso p  
aralelo 201512\veh':

**Opciones Grid:**

A continuación se mencionan las modificaciones a realizar sobre los programas con el fin de utilizar funciones Grid.

Líneas Generales:

Todos los programas que componen el proceso, deberán tener añadida las siguientes líneas al inicio:

Activación servicio Grid:

```
%let rc=%sysfunc(grdsvc_enable(_all_,server=SASApp));
```

```
options autologon;
```

Variables adicionales para ejecución Jobs Grid:

```
options emailhost=("172.29.64.20" port=25);
```

```
%let mailaddr= S40073@CREDITO.BCP.COM.PE;
```

Definición de Jobs Grid:

Se detallan los Jobs Grid identificados para el proceso.

03\_MinoristaConsumo\_PARAM\_NEW.sas:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_Cons:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;  
tempdate=put(date(),ddmmyyn8.);  
temptime=put(time(),b8601tm.);  
call symput('currdate',catx('_',tempdate,temptime));  
run;
```

/\*Definición del Job\*/

```
rsubmit job_Cons wait=no cmacvar=job_Cons sysrputsync=yes  
log="/sasdata01/Soluciones_Informacion/LT3_BaseParametros/logs/Minorista  
Consumo_job_&currdate..log" new  
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio  
Ejecución Base parámetros – Minorista Consumo Job_Cons ";  
;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:  
endrssubmit;

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.  
waitfor \_all\_ job\_Cons;

03\_MinoristaHipotecario\_PARAM\_NEW\_lgd\_octubre.sas:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_MHip:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

/\*Inicializar fecha de comienzo del job\*/

```
data _null_;  
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;
```

/\*Definición del Job\*/

```
rsubmit job_MHip wait=no cmacvar=job_MHip sysrputsync=yes
log="/sasdata01/Soluciones_Informacion/LT3_BaseParametros/logs/Minorista
aHipotecario_job_&currdate..log" new
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio
Ejecución Base parámetros – Minorista Hipotecario Job_MHip ";
;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_Hip;
```

03\_MinoristaPymeNoRev\_PARAM\_NEW.sas:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_PyNRv:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

/\*Definición del Job\*/

```
rsubmit job_PyNRv wait=no cmacvar=job_PyNRv sysrputsync=yes  
log="/sasdata01/Soluciones_Informacion/LT3_BaseParametros/logs/Minorist  
aPymeNoRev_job_&currdate..log" new  
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio  
Ejecución Base parámetros – Minorista Pyme No rev Job_PyNRv ";  
;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_PyNRv;
```

03\_MinoristaPymeRev\_PARAM\_NEW.sas:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_PyRv:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

```
/*Definición del Job*/
```

```
rsubmit job_PyRv wait=no cmacvar=job_PyRv sysrputsync=yes
```

```
log="/sasdata01/Soluciones_Informacion/LT3_BaseParametros/logs/MinoristaPymRev_job_&currdate..log" new
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio
Ejecución Base parámetros – Minorista Pyme Rev Job_PyRv ";
;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_PyRv;
```

03\_MinoristaTarjeta2\_PARAM\_NEW.sas:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_MTC:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

```
/*Definición del Job*/
```

```
rsubmit job_MTC wait=no cmacvar=job_MTC sysrputsync=yes
```

```
log="/sasdata01/Soluciones_Informacion/LT3_BaseParametros/logs/MinoristaTarjetas_job_&currdate..log" new
```

```
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio
```

```
Ejecución Base parámetros – Minorista tarjetas Job_MTC ";
```

```
;
```



Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_MTC;
```

03\_MinoristaVehicular2\_PARAM\_NEW.sas:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_MVh:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/  
data _null_;  
tempdate=put(date(),ddmmyyn8.);  
temptime=put(time(),b8601tm.);  
call symput('currdate',catx('_',tempdate,temptime));  
run;
```

/\*Definición del Job\*/

```
rsubmit job_MVh wait=no cmacvar=job_MVh sysrputsync=yes  
log="/sasdata01/Soluciones_Informacion/LT3_BaseParametros/logs/MinoristaVehicular_job_&currdate..log" new  
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio  
Ejecución Base parámetros – Minorista Vehicular Job_MVh ";  
;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_MVh;
```



**ANEXO 10**  
**CONSIDERACIONES PARA LA MIGRACIÓN DEL PROCESOS SAS**  
**SEGUIMIENTO\_CONTINUO DEL ÁREA DE SOLUCIONES DE**  
**INFORMACIÓN**

**Estructura del proceso migrado:**

Se reutilizará el proyecto Guide existente, editando los programas correspondientes.

Consideraciones para la migración del proceso:

A continuación se mencionan con el máximo nivel de detalle posible las consideraciones y/o modificaciones a realizar sobre las fuentes del proceso:

Precondición:

Verificar que se encuentre configuradas las librerías Oracle, que son utilizadas como Input del proceso.

Verificar que se encuentre creada la estructura de directorios necesaria para el proceso.

**Modificaciones sobre el proyecto Guide:**

Modificar los programas:

0. Preparar Macros:

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

LIBNAME MACROS

"\\Psasp04\SI\_ADR\_PROD\03\_PROCESOS\SEGUIMIENTO\Macros";

PROC IMPORT

DATAFILE=\\Psasp04\SI\_ADR\_PROD\03\_PROCESOS\SEGUIMIENTO\Par  
ametros\_CCO.CSV

%LET

V DIR=%SYSFUNC(CATX(\,%SYSFUNC(COMPRESS(&V R RESULT.,'  
')),%SYSFUNC(COMPRESS(&V M ANALISIS.,''))));

1. Import Oracle to SAS:

No se realizaran modificaciones.

. Cat Var Admision CCO:

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

LIBNAME MACROS

"\\Psasp04\SI ADR PROD\03 PROCESOS\SEGUIMIENTO\Macros";

PROC IMPORT

DATAFILE=\\Psasp04\SI ADR PROD\03 PROCESOS\SEGUIMIENTO\Par  
ametros\_CCO.CSV

%LET

V DIR=%SYSFUNC(CATX(\,%SYSFUNC(COMPRESS(&V R RESULT.,'  
')),%SYSFUNC(COMPRESS(&V M ANALISIS.,''))));

1. Cat Var - Port Tot CCO:

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

LIBNAME MACROS

"\\Psasp04\SI ADR PROD\03 PROCESOS\SEGUIMIENTO\Macros";

PROC IMPORT

DATAFILE=\\Psasp04\SI ADR PROD\03 PROCESOS\SEGUIMIENTO\Par  
ametros\_CCO.CSV

%LET

V DIR=%SYSFUNC(CATX(\,%SYSFUNC(COMPRESS(&V R RESULT.,'  
')),%SYSFUNC(COMPRESS(&V M ANALISIS.,''))));

2. Cat Var - Port Tempt CCO:

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

LIBNAME MACROS

"\\Psasp04\SI ADR PROD\03 PROCESOS\SEGUIMIENTO\Macros";

PROC IMPORT

DATAFILE=\\Psasp04\SI ADR PROD\03 PROCESOS\SEGUIMIENTO\Par  
ametros\_CCO.CSV

%LET

V DIR=%SYSFUNC(CATX(\,%SYSFUNC(COMPRESS(&V R RESULT.,'  
')),%SYSFUNC(COMPRESS(&V M ANALISIS.,''))));

5. Tableros Adm:

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

LIBNAME MACROS

"\\Psasp04\SI\_ADR\_PROD\03\_PROCESOS\SEGUIMIENTO\Macros";

PROC IMPORT

DATAFILE=\\Psasp04\SI\_ADR\_PROD\03\_PROCESOS\SEGUIMIENTO\Par  
ametros\_CCO.CSV

%LET

V DIR=%SYSFUNC(CATX(\,%SYSFUNC(COMPRESS(&V R RESULT.,'  
')),%SYSFUNC(COMPRESS(&V M ANALISIS.,''))));

PROC IMPORT

DATAFILE="&VAR RUTA DATOS.\DRIVERS\_CCO\_ADM.CSV"

PROC IMPORT

DATAFILE="&VAR RUTA DATOS.\DRIVERS\_CCO\_COMP.CSV"

FILENAME MYFILE "&VAR RUTA DATOS.\LOG\_SEG\_CCO.LOG";

5. Tableros Comp:

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

LIBNAME MACROS

"\\Psasp04\SI\_ADR\_PROD\03\_PROCESOS\SEGUIMIENTO\Macros";

PROC IMPORT

DATAFILE=\\Psasp04\SI\_ADR\_PROD\03\_PROCESOS\SEGUIMIENTO\Par  
ametros\_CCO.CSV

%LET

V DIR=%SYSFUNC(CATX(\,%SYSFUNC(COMPRESS(&V R RESULT.,'  
')),%SYSFUNC(COMPRESS(&V M ANALISIS.,''))));

### Opciones Grid:

A continuación se mencionan las modificaciones a realizar sobre los programas con el fin de utilizar funciones Grid.

Líneas Generales:

Todos los programas que componen el proceso, deberán tener añadida las siguientes líneas al inicio:

Activación servicio Grid:

```
%let rc=%sysfunc(grdsvc_enable(_all_,server=SASApp));  
options autosignon;
```

Variables adicionales para ejecución Jobs Grid:

```
options emailhost=("172.29.64.20" port=25);  
%let mailaddr= S40073@CREDITO.BCP.COM.PE;
```

Definición de Jobs Grid:

Se detallan los Jobs Grid identificados para el proceso.

1. Import\_Oracle\_to\_SAS:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_Imp:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/  
data _null_;  
tempdate=put(date(),ddmmyyn8.);  
temptime=put(time(),b8601tm.);  
call symput('currdate',catx('_',tempdate,temptime));  
run;
```

/\*Definición del Job\*/

```
rsubmit job_Imp wait=no cmacvar=job_Imp sysrputsync=yes  
inheritlib=(INPUT)  
log="/sasdata01/Soluciones_Informacion/LT3_SeguimientoContinuo/logs/Consumo_job_&currdate..log" new  
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio  
Ejecución Seguimiento Continuo–Consumo Job_Imp ";;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_imp;
```

Cat Var Admision CCO, 1. Cat Var - Port Tot CCO, 2. Cat Var - Port Tempt CCO:

Se crearán 3 jobs que se ejecutarán en paralelo. Cada uno de los Jobs incluirá el código de los 3 programas mencionados.

Job\_CatV1:

Incluirá todo el código de . Cat Var Admision CCO

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

```
/*Definición del Job*/
```

```
rsubmit job_CatV1 wait=no cmacvar=job_CatV1 sysrputsync=yes
```

```
inheritlib=(MACROS )
```

```
log="/sasdata01/Soluciones_Informacion/LT3_SeguimientoContinuo/logs/Co
```

```
nsumo_job_CatV1_&currdate..log" new
```

```
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio
```

```
Ejecución Seguimiento Continuo–Consumo Job_CatV1";
```

```
;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Job\_CatV2:

Incluirá todo el código de 1. Cat Var - Port Tot CCO

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

```
/*Definición del Job*/
```

```
rsubmit job_CatV2 wait=no cmacvar=job_CatV2 sysrputsync=yes
```

```
inheritlib=(MACROS )
```

```
log="/sasdata01/Soluciones_Informacion/LT3_SeguimientoContinuo/logs/Consumo_job_CatV2_&currdate..log" new
```

```
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio
```

```
Ejecución Seguimiento Continuo–Consumo Job_CatV2";
```

```
;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Job\_CatV3:

Incluirá todo el código de 2. Cat Var - Port Tempt CCO

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:



```

/*Inicializar fecha de comienzo del job*/
data _null_;
tempdate=put(date(),ddmmyyn8.);
temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;

```

```

/*Definición del Job*/
rsubmit job_CatV3 wait=no cmacvar=job_CatV3 sysrputsync=yes
inheritlib=(MACROS )
log="/sasdata01/Soluciones_Informacion/LT3_SeguimientoContinuo/logs/Co
nsumo_job_CatV3_&currdate..log" new
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio
Ejecución Seguimiento Continuo–Consumo Job_CatV3";
;

```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución de Job\_CatV1, Job\_CatV2, Job\_CatV3:

Agregar la siguiente línea al final del código para ejecutar en paralelo los 3 jobs

```
waitfor _all_ Job_CatV1, Job_CatV2, Job_CatV3;
```

4. Dummy:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_Dummy:

Comienzo del Jobs:

Añadir las siguientes líneas:

```

/*Inicializar fecha de comienzo del job*/
data _null_;
tempdate=put(date(),ddmmyyn8.);

```

```
temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;
```

/\*Definición del Job\*/

```
rsubmit job_Dummy wait=no cmacvar=job_Dummy sysrputsync=yes
inheritlib=(INPUT)
log="/sasdata01/Soluciones_Informacion/LT3_SeguimientoContinuo/logs/Co
nsumo_job_Dummy_&currdate..log" new
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio
Ejecución Seguimiento Continuo–Consumo Job_Dummy ";
;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_Dummy;
```

5. Tableros Adm y 5. Tableros Comp:

Se crearán 2 jobs que se ejecutarán en paralelo.

Cada uno de los Jobs incluirá el código de los 3 programas mencionados.

Job\_TAdm:

Incluirá todo el código de 5. Tableros Adm

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));  
run;
```

```
/*Definición del Job*/
```

```
rsubmit job_TAdm wait=no cmacvar=job_TADM sysrputsync=yes  
log="/sasdata01/Soluciones_Informacion/LT3_SeguimientoContinuo/logs/Co  
nsumo_job_TAdm_&currdate..log" new  
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio  
Ejecución Seguimiento Continuo–Consumo Job_TAdm";  
;
```

```
Fin del job:
```

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

```
Job_TComp:
```

Incluirá todo el código de 5. Tableros Adm

```
Comienzo del Jobs:
```

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Inicializar fecha de comienzo del job*/  
data _null_;  
tempdate=put(date(),ddmmyyn8.);  
temptime=put(time(),b8601tm.);  
call symput('currdate',catx('_',tempdate,temptime));  
run;
```

```
/*Definición del Job*/
```

```
rsubmit job_TComp wait=no cmacvar=job_TComp sysrputsync=yes  
log="/sasdata01/Soluciones_Informacion/LT3_SeguimientoContinuo/logs/Co  
nsumo_job_TComp_&currdate..log" new  
notify="&mailaddr" subject=" Notificación desde &syshostname: Inicio  
Ejecución Seguimiento Continuo–Consumo Job_TComp";  
;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrssubmit;
```

Ejecución de Job\_TAdm, Job\_TComp

Agregar la siguiente línea al final del código para ejecutar en paralelo los 2 jobs

```
waitfor _all_ Job_TAdm Job_TComp;
```



## ANEXO 11

### CONSIDERACIONES PARA LA MIGRACIÓN DEL PROCESOS SAS BEHAVIOR\_PYME DEL ÁREA DE SOLUCIONES DE INFORMACIÓN

#### Estructura del proceso migrado:

Se generará un proyecto Guide con 6 flujos diferentes, uno para cada subproceso.

Cada uno de los flujos tendrá el programa SAS correspondiente.

Consideraciones para la migración del proceso:

A continuación se mencionan con el máximo nivel de detalle posible las consideraciones y/o modificaciones a realizar sobre las fuentes del proceso:

Precondición:

Verificar que se encuentre configuradas las librerías Oracle, que son utilizadas como Input del proceso.

Verificar que se encuentre creada la estructura de directorios necesaria para el proceso.

#### Modificaciones sobre el proyecto Guide:

Modificar los programas:

##### 0.Parametros:

Crear un nuevo parámetro MES y solicitarlo al iniciar la ejecución.

Modificar los directorios correspondientes a librerías y cambiado “\” por “/”.

##### LIBNAME UNIV BASE

"\\psasp04\SI ADR DESA\02 Proyectos\201601 Behavior Pyme Cliente\Procesamiento";

...

##### LIBNAME RES\_SAL

"\\PSASP04\SI ADR MV 01\00 MatrizVariables\02 Resumen Saldo\Procesados";

##### LIBNAME VP\_1

"\\PSASP04\SI ADR MV 01\00 MatrizVariables\06 Facturacion VP\_1\Procesados";

## LIBNAME VP\_2

"\\PSASP04\SI ADR MV 01\00 MatrizVariables\03 Facturacion VP\_2\Procesados";

## LIBNAME EXP\_BCP

"\\psasp04\SI ADR MV 02\00.Laboratorio\201407 MatrizModelamiento\05 ExperienciaBCP\Procesados";

## LIBNAME RCC

"\\psasp04\SI ADR MV 02\00.Laboratorio\201407 MatrizModelamiento\08 Nuevas Var RCC\Procesamiento";

### 1.Universo Clientes Pyme:

No se realizaran modificaciones.

#### 2.1.Modulo Solicitudes:

No se realizaran modificaciones.

#### 2.2.Modulo Activos:

No se realizaran modificaciones.

#### 2.3.Modulo Pasivos

No se realizaran modificaciones.

#### 2.4.Modulo RCC

No se realizaran modificaciones.

### 3.Matriz Implementacion

No se realizaran modificaciones.

### 5.Calibracion Cliente

No se realizaran modificaciones.

### 6.Calibracion Producto

No se realizaran modificaciones.

### 7.Export SAS a Oracle

Se eliminará el seteo de la librería PROY\_RBP, agregándose este seteo en el paso 0. Parámetros.

libname PROY\_RBP ORACLE user=PROY\_RBP pass='GIS7532159'  
path=BCPDW3 schema=PROY\_RBP;

## **Opciones Grid:**

A continuación se mencionan las modificaciones a realizar sobre los programas con el fin de utilizar funciones Grid.

### Líneas Generales:

Todos los programas que componen el proceso, deberán tener añadida las siguientes líneas al inicio:

Activación servicio Grid:

```
%let rc=%sysfunc(grdsvc_enable(_all_,server=SASApp));  
options autosignon;
```

Variables adicionales para ejecución Jobs Grid:

```
options emailhost=("172.29.64.20" port=25);  
%let mailaddr= S40073@CREDITO.BCP.COM.PE;
```

Definición de Jobs Grid:

Se detallan los Jobs Grid identificados para el proceso.

0.Parametros:

No se agregará Código Grid en este paso.

1.Universo\_Clientes\_Pyme:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_UPyme:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/  
%syslput MES=&MES. /remote=job_UPyme;
```

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;  
tempdate=put(date(),ddmmyyn8.);  
temptime=put(time(),b8601tm.);  
call symput('currdate',catx('_',tempdate,temptime));  
run;
```

Parametros

```
/*Definición del Job*/
```

```

rsubmit job_UPyme wait=no cmacvar=job_UPyme sysrputsync=yes
inheritlib=(UNIV PROY_RBP)
log="/sasdata01/Soluciones_Informacion/LT3_Behavior_Pyme/logs/Universo
_Clientes_Pyme_job_UPyme_&currdate..log" new
notify="&mailaddr" subject=" Notification desde &syshostname: Inicio
Ejecución Behavior Pyme–Universo CLientes Pyme Job_UPyme";
;

```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

La siguiente línea tiene por objetivo realizar la ejecución del job definido, y en caso que tenga dependencias aguardará o paralelizará las ejecuciones.

```
waitfor _all_ job_UPyme;
```

2.1.Modulo\_Solicitudes:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_MSol:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```

/*Incluir variables que se utilizan dentro del job*/
%syslput MES=&MES. /remote=job_MSol;
%syslput MES_1=&MES_1. /remote=job_MSol;
/*Inicializar fecha de comienzo del job*/
data _null_;
tempdate=put(date(),ddmmyyn8.);
temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;

```



## Parámetros

/\*Definición del Job\*/

```
rsubmit job_MSol wait=no cmacvar=job_MSol sysrputsync=yes  
inheritlib=(UNIV PROY_RBP)  
log="/sasdata01/Soluciones_Informacion/LT3_Behavior_Pyme/logs/Modulo_  
Solicitudes_job_MSol_&currdate..log" new  
notify="&mailaddr" subject=" Notification desde &syshostname: Inicio  
Ejecucion Behavior Pyme-Modulo Solicitudes Job_MSol";  
;
```

## Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

## Ejecución del job:

Definido en el siguiente paso, ya que se ejecutará en paralelo.

### 2.2.Modulo\_Activos:

Se creará un nuevo Job, que incluirá todo el proceso.

## Job\_MAct:

### Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/  
%syslput MES=&MES. /remote=job_MAct;  
%syslput MES_1=&MES_1. /remote=job_MAct;  
/*Inicializar fecha de comienzo del job*/  
data _null_;  
tempdate=put(date(),ddmmyyn8.);  
temptime=put(time(),b8601tm.);  
call symput('currdate',catx('_',tempdate,temptime));  
run;
```

## Parámetros

/\*Definición del Job\*/

```
rsubmit job_MAct wait=no cmacvar=job_Mact sysrputsync=yes
```

```
inheritlib=(UNIV PROY_RBP)
```

```
log="/sasdata01/Soluciones_Informacion/LT3_Behavior_Pyme/logs/Modulo_
Activos_job_MAct_&currdate..log" new
```

```
notify="&mailaddr" subject=" Notification desde &syshostname: Inicio
Ejecucion Behavior Pyme-Modulo Activos Job_MAct";
```

```
;
```

## Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

## Ejecución del job:

Definido en el siguiente paso, ya que se ejecutará en paralelo.

Módulo\_Pasivos creará un nuevo Job, que incluirá todo el proceso.

Job\_MPas:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/
```

```
%syslput MES=&MES. /remote=job_MPas;
```

```
%syslput MES_1=&MES_1. /remote=job_MPas;
```

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

## Parametros

/\*Definición del Job\*/

```
rsubmit job_MPas wait=no cmacvar=job_MPas sysrputsync=yes
```

```
inheritlib=(UNIV PROY_RBP)
```

```
log="/sasdata01/Soluciones_Informacion/LT3_Behavior_Pyme/logs/Modulo_Pasivos_job_MPas_&currdate..log" new
```

```
notify="&mailaddr" subject=" Notification desde &syshostname: Inicio Ejecucion Behavior Pyme-Modulo Pasivos Job_MPas";
```

```
;
```

## Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

## Ejecución del job:

Definido en el siguiente paso, ya que se ejecutará en paralelo.

### 2.4.Modulo\_RCC:

Se creará un nuevo Job, que incluirá todo el proceso.

#### Job\_MRCC:

#### Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/
```

```
%syslput MES=&MES. /remote=job_MRCC;
```

```
%syslput MES_1=&MES_1. /remote=job_MRCC;
```

```
/*Inicializar fecha de comienzo del job*/
```

```
data _null_;
```

```
tempdate=put(date(),ddmmyyn8.);
```

```
temptime=put(time(),b8601tm.);
```

```
call symput('currdate',catx('_',tempdate,temptime));
```

```
run;
```

## Parámetros

/\*Definición del Job\*/

```
rsubmit job_MRCC wait=no cmacvar=job_MRCC sysrputsync=yes  
inheritlib=(UNIV PROY_RBP)  
log="/sasdata01/Soluciones_Informacion/LT3_Behavior_Pyme/logs/Modulo_  
RCC_job_MRCC_&currdate..log" new  
notify="&mailaddr" subject=" Notification desde &syshostname: Inicio  
Ejecucion Behavior Pyme-Modulo RCC Job_MRCC";  
;
```

## Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

## Ejecución del job:

Definido en el siguiente paso, ya que se ejecutará en paralelo.

## Paralelización jobs:

2.1.Modulo\_Solicitudes

2.2.Modulo\_Activos

2.3.Modulo\_Pasivos

2.4.Modulo\_RCC

Agregar en un nuevo programa, o al finalizar las definiciones de los Jobs, la siguiente línea que indica la ejecución en paralelo de los Jobs.

```
waitfor _all_ Job_MSol Job_MAct Job_MPas Job_MRCC;
```

## 3.Matriz\_Implementacion:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_MImp:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/
```

```
%syslput MES=&MES. /remote=job_MImp;
```

```

/*Inicializar fecha de comienzo del job*/
data _null_;
    tempdate=put(date(),ddmmyyn8.);
    temptime=put(time(),b8601tm.);
    call symput('currdate',catx('_',tempdate,temptime));
run;

```

Parámetros

```

/*Definición del Job*/

```

```

rsubmit job_MImp wait=no cmacvar=job_MImp sysrputsync=yes
inheritlib=(UNIV)
log="/sasdata01/Soluciones_Informacion/LT3_Behavior_Pyme/logs/Matriz_I
mplementación_job_MImp_&currdate..log" new
notify="&mailaddr" subject=" Notification desde &syshostname: Inicio
Ejecucion Behavior Pyme–Matriz Implementacion Job_MImp";
;

```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```

endrsubmit;

```

Ejecución del job:

```

waitfor _all_ Job_MImp;

```

4.TU\_xb\_combinado:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_XbCom:

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```

/*Incluir variables que se utilizan dentro del job*/

```

```

%syslput MES=&MES. /remote=job_XbCom;

```

```

%syslput mis_list =&mis_list. /remote=job_XbCom;

```

```

/*Inicializar fecha de comienzo del job*/
data _null_;
    tempdate=put(date(),ddmmyyn8.);
    temptime=put(time(),b8601tm.);
    call symput('currdate',catx('_',tempdate,temptime));
run;

```

#### Parámetros

```

/*Definición del Job*/

```

```

rsubmit job_XbCom wait=no cmacvar=job_XbCom sysrputsync=yes
inheritlib=(UNIV)
log="/sasdata01/Soluciones_Informacion/LT3_Behavior_Pyme/logs/TU_xb_
Combinado_job_XbCom_&currdate..log" new
notify="&mailaddr" subject=" Notification desde &syshostname: Inicio
Ejecucion Behavior Pyme–TU xb Combinado Job_XbCom";
;

```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```

endrsubmit;

```

Ejecución del job:

```

waitfor _all_ Job_XbCom;

```

#### 5. Calibración\_Cliente:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_CCli:

Precondición: Se requiere que la tabla importada vía Excel, en vez de generarse en work, se genere en la librería Univ.

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```

/*Incluir variables que se utilizan dentro del job*/

```

```

%syslput MES=&MES. /remote=job_CCli;

```

```

/*Inicializar fecha de comienzo del job*/
data _null_;
tempdate=put(date(),ddmmyyn8.);
temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;

```

#### Parámetros

```
/*Definición del Job*/
```

```

rsubmit job_CCli wait=no cmacvar=job_CCli sysrputsync=yes
inheritlib=(UNIV PROY_RBP)
log="/sasdata01/Soluciones_Informacion/LT3_Behavior_Pyme/logs/Calibracion_Cliente_job_CCli_&currdate..log" new
notify="&mailaddr" subject=" Notification desde &syshostname: Inicio
Ejecucion Behavior Pyme-Calibración Cliente Job_CCli";
;

```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

```
waitfor _all_ Job_CCli;
```

6.Calibracion\_Producto:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_CPro:

Precondición: Se requiere que la tabla importada vía Excel, en vez de generarse en work, se genere en la librería Univ.

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/
```

```
%syslput MES=&MES. /remote=job_Cpro;
```

```

/*Inicializar fecha de comienzo del job*/
data _null_;
tempdate=put(date(),ddmmyyn8.);
temptime=put(time(),b8601tm.);
call symput('currdate',catx('_',tempdate,temptime));
run;

```

Parámetros

```
/*Definición del Job*/
```

```

rsubmit job_CPro wait=no cmacvar=job_CPro sysrputsync=yes
inheritlib=(UNIV PROY_RBP)
log="/sasdata01/Soluciones_Informacion/LT3_Behavior_Pyme/logs/Calibracion_Producto_job_CPro_&currdate..log" new
notify="&mailaddr" subject=" Notification desde &syshostname: Inicio
Ejecucion Behavior Pyme-Calibración Producto Job_CPro";
;

```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;
```

Ejecución del job:

```
waitfor _all_ Job_CPro;
```

7.Export\_SAS\_a\_Oracle:

Se creará un nuevo Job, que incluirá todo el proceso.

Job\_Exp:

Precondición: Se requiere que la tabla importada vía Excel, en vez de generarse en work, se genere en la librería Univ.

Comienzo del Jobs:

Previo a realizar la consulta Oracle, añadir las siguientes líneas:

```
/*Incluir variables que se utilizan dentro del job*/
```

```
%syslput MES=&MES. /remote=job_Exp;
```



```
/*Inicializar fecha de comienzo del job*/  
data _null_;  
tempdate=put(date(),ddmmyyn8.);  
temptime=put(time(),b8601tm.);  
call symput('currdate',catx('_',tempdate,temptime));  
run;
```

Parámetros

```
/*Definición del Job*/
```

```
rsubmit job_Exp wait=no cmacvar=job_Exp sysrputsync=yes inheritlib=(UNIV  
PROY_RBP)  
log="/sasdata01/Soluciones_Informacion/LT3_Behavior_Pyme/logs/Export_T  
o_Oracel_job_Exp_&currdate..log" new  
notify="&mailaddr" subject=" Notification desde &syshostname: Inicio  
Ejecucion Behavior Pyme-Export to Oracle Job_Exp";  
;
```

Fin del job:

La siguiente línea, deberá ser añadida al final del programa para identificar la finalización del job:

```
endrsubmit;  
Ejecución del job:  
waitfor _all_ Job_Exp;
```