

A B G

APLICATIVO BANCARIO GENERICO

Aprendiendo a Programar

Aprendamos a Programar con ABG

En base a las experiencias utilizadas, por muchos técnicos, para el aprendizaje de nuevos lenguajes o entornos de desarrollo, es que vamos a encarar un esquema de adquirir conocimiento basado en preguntas elementales de programación y como las mismas son resueltas con el lenguaje de programación de ABG, para la plataforma IBM AS400.

Esta metodología condiciona la existencia de un conocimiento previo en el desarrollo de sistemas informáticos a través de algún lenguaje de programación.

Nos basaremos en una enumeración de preguntas puntuales, con la explicación que corresponda y la codificación necesaria para responder a lo solicitado.

1. Que es un Programa ABG.

- a. El Aplicativo ABG fue diseñado para soportar de manera integrada, las distintas necesidades de sistematización de los sectores de una Institución Financiera (y también Comercial, Productiva, etc.). Esto se logra a través de la definición, personalización y gestión básica de los Productos que se necesiten sistematizar. Para satisfacer las necesidades particulares de gestión, se debe utilizar el entorno de desarrollo ABG, el cual permite crear los distintos programas necesarios.
- b. Para crear un programa ABG, primeramente hay que definir y personalizar una Transacción, considerando para ello:
 - i. La Familia de Producto a la cual pertenecerá.
 - ii. La habilitación de las variables básicas que se utilizarán (Fechas, Comprobantes, Importes, Codificaciones, Tasas/Coeficientes/Valores).
 - iii. Los Eventos que se codificarán para resolver la necesidad operativa.
- c. Los Eventos de una transacción son codificaciones agrupadas por funcionalidad, con el objetivo de organizar y normalizar la codificación necesaria de las transacciones. La utilización de los eventos se realiza de la siguiente manera:
 - i. El Evento 0000 (Ingreso) permite visualizar, solicitar y validar los datos necesarios para el procesamiento.
 - ii. El Evento 9999 (Actualizar) permite definir la codificación necesaria para la afectación de las tablas operativas.
 - iii. El Evento 9998 (Contabilizar) permite definir los pares contables necesarios para efectuar los registros contables necesarios.
 - iv. El Evento 9997 (Cálculos Definibles) Posibilita la definición de los criterios de aplicación y la expresiones de valuaciones.
 - v. Desde el Evento 0001 hasta el 9996 se utilizan para desarrollar los propósitos generales.
- d. Por lo tanto, en base a lo anteriormente descripto, podemos decir que un programa ABG es un Evento de una Transacción ABG. Esta empaqueta a los eventos ejecutables que la constituyen. Por ejemplo en una Transacción monetaria se requiere solicitar datos (codificados en el evento 0000), dichos datos se procesan (evento 0001), se impactan en las bases operativas (evento 9999), se contabilizan (evento 9998). Como auxiliar al proceso puede ser necesario que se realicen cálculos definibles (evento 9997).

2. Como se invocan los Programas ABG.

- a. El aplicativo ABG dispone de una interfaz que permite invocar las Transacciones o Programas ABG, dentro de un marco controlado por el esquema de seguridad del aplicativo.
 - b. Una vez que se cumplieron con los controles de seguridad, se procede a la invocación del evento 0000 (ingreso), el cual debe prever el requerimiento de los datos necesarios, aplicando los controles que correspondan y en caso de que los mismos sean correctos, corresponde que se invoquen a los eventos que fueron codificados para satisfacer la necesidad operativa.
 - c. Otra forma de invocar a una Transacción/Evento o Programa ABG, consiste en activar al runtime especificando la identidad de la Transacción/Evento y los valores de los parámetros en caso de ser necesarios.
3. Como se crean las nuevas tablas que pueden necesitarse en una aplicación codificada con ABG.
- a. Existen distintas formas:
 - i. Mediante la especificación de la estructura a través de los miembros fuentes de definición de datos (DDS) y la utilización del comando CRTPF o CRTLF dentro de la sentencia **Exec “CrtPf File(Biblioteca/ArchDatos) Srcfile(Biblioteca/ArchFuente) SrcMbr(*File)”**, que es la que se encarga de ejecutar un comando del Sistema Operativo. Si la ejecución produce algún error la variable **&HDRTRN** tendrá valor **1076**, asignando el código de error del sistema operativo en la variable **&ERRMSG** y la descripción del mismo en la variable **&ERRTXT**.
 - ii. Utilizando el entorno SQL y dentro del mismo, a través del comando CREATE TABLE, CREATE INDEX o CREATE VIEW.
 - iii. También podemos utilizar la sentencia de ABG **SqlExc “Sentencia SQL”**, la cual permite ejecutar un comando SQL (CREATE TABLE, CREATE INDEX o CREATE VIEW). Si la ejecución produce algún error la variable **&HDRTRN** tendrá valor **1077**, asignando el código de error del sistema operativo en la variable **&ERRMSG** y la descripción del mismo en la variable **&ERRTXT**.
4. Como se define una tabla que se necesita utilizar en un programa ABG.
- a. Las tablas no se definen en los programas ABG, solamente se utilizan.
5. Como se acceden a las filas o registros de una tabla en un programa ABG.
- a. Como el lenguaje ABG fue desarrollado para soportar las necesidades particulares de los Productos definidos y personalizados en el Aplicativo ABG, es necesario destacar que existe un conjunto de tablas para soportan al módulo base, donde algunas de ellas son accedidas mediante sentencias específicas:
 - i. **LeeReg** la cual permite leer un registro de una tabla predefinida.
 - ii. **WrtReg** mediante la cual podemos grabar los datos del registro activo de una tabla especificada.
 - iii. **UpdReg**, mediante la cual podemos modificar los datos del registro activo de una tabla especificada.
 - iv. **LeeSdo**, permite acceder a un registro de Saldo Operativo de una Cuenta en un Producto habilitado para la misma (tabla predefinida).
 - b. Para las otras tablas solamente se acceden a través de sentencias SQL.
 - i. **SqlLee**, la cual permite especificar el comando SELECT de SQL para recuperar determinadas columnas de una fila o registro de una o varias tablas.

- ii. **SqlCalc**, permite recuperar solamente una función SQL (SUM, MAX, MIN, AVG) de una columna numérica perteneciente a una determinada tabla.
 - iii. **SqlExc**, permite ejecutar inmediatamente un comando SQL (CREATE TABLE, CREATE INDEX, CREATE VIEW, INSERT, UPDATE, DELETE, etc.)
 - iv. **SqlSlt** utilizado para acceder a un conjunto de registros de una o varias tablas, con el propósito de satisfacer las necesidades de funcionalidad de las siguiente sentencias ABG:
 1. **Report**, generar un reporte.
 2. **GenTxt**, generar un archivo dentro de una carpeta.
 3. **LeeGrp**, procesar un conjunto de registros o filas.
 4. **SflExc**, paginar los registros o filas de una o varias tablas.
6. Donde se almacenan los datos de las columnas de los registros o filas recuperadas.
 - a. Los datos son almacenados en las variables ABG, para las cuales existen distintas categorías:
 - i. Globales, corresponde a todas las variables pertenecientes a las tablas bases (predefinidas) del aplicativo y que se definen automáticamente en toda Transacción/Evento o Programa ABG. El valor de estas variables es mantenido durante toda la sesión ABG, es decir que se mantiene para todas las transacciones o programas que se vayan invocando.
 - ii. Locales, son aquellas que corresponden definirse en cada Transacción/Evento o Programa ABG que las necesite, las cuales solamente pueden pasarse como parámetro en la invocación de otra Transacción/Evento o Programa ABG.
 - iii. Parámetros, son aquellas que reciben los valores de los parámetros de entradas en una Transacción/Evento o Programa ABG. Estas también son consideradas como locales.
7. Es posible definir estructuras en la programación ABG.
 - a. La sentencia **DefVar** permite definir una variable que puede ser considerada como estructura, mediante la especificación de una tabla referenciada, desde la cual se copiaran las variables o campos que la constituyen.
 - b. También es posible efectuar redefiniciones de variable a través de la especificación de parámetros optativos en la sentencia **DefVar**.
8. Los vectores son soportados por ABG.
 - a. La sentencia **DefVar** también permite definir que una variable sea considerada como un vector, esto se logra con la especificación de un parámetro que lo define como tal.
9. Que sentencias de control de flujo existen en ABG.
 - a. Existen todas las sentencias elementales para el control de flujo:
 - i. **If, And, Or, Else, EndIf**.
 - ii. **Do, EndDo**.
 - iii. **DoWhile, EndDw**.
 - iv. **DoUntil, EndDu**.
 - v. **Select, When, Other, EndDh**.
10. Como se hace para producir una salida por impresora.
 - a. Existen tres formas:

- i. Mediante la sentencia **Report**, la cual requiere que previamente se realice el diseño del listado, utilizando para ello el asistente correspondiente.
- ii. Utilizando la sentencia **PrtFor**, también requiere que previamente, utilizando el asistente, se realice el diseño del formulario.
- iii. A través del uso de la sentencia **PrtLin**, se pueden construir listados o formularios sin la necesidad de diseñarlos. En este caso corresponde utilizar la sentencia **OpnPrt** para activar la impresora y **CloPrt** para finalizar la impresión.

11. Como se diseñan y/o utilizan los formatos de pantallas en ABG.

- a. En ABG se dispone de un conjunto de formatos de pantallas predefinidos, a través de los cuales se satisfacen la mayoría de las necesidades de sistematización, para el caso de que ningún formato satisfaga la necesidad, es posible construir formatos, armados dinámicamente en ejecución con el objetivo de requerir y/o visualizar la información correspondiente en cada caso. Existe la posibilidad de que se utilicen múltifichas
- b. Las sentencias ABG que permiten trabajar o que están asociadas con los formatos de pantalla (predefinidos o definibles) son:
 - i. ExeFmt, permite requerir los datos para un formato.
 - ii. WrtFmt, se utiliza para visualizar los datos en un formato.
 - iii. ExeMsg, visualiza un mensaje y solicita una confirmación.
 - iv. WrtMsg, visualiza un mensaje.
 - v. InzPanVar, inicializa las áreas que son utilizadas para la construcción de un formato definible, a partir de campos especificados.
 - vi. FldPanVar, especifican los campos (conjuntamente con los atributos necesarios) que serán utilizados para la construcción de un formato definibles.
 - vii. TxtSolapas, permite especificar que el formato definible que se va a utilizar, tendrá distintas solapas de información, esto se puede utilizar cuando los datos visualizar y/o ingresar son mucho mas que lo permitido por el mayor formato definible o por que existe una necesidad de agrupación.
 - viii. TxtBotones, permite especificar que el formato definible que se va a utilizar, tendrá botones para seleccionar acciones.
 - ix. Descrboton, permite especificar la descripción ampliada que le corresponde a cada botón que se utilice.
 - x. MnuOpc, permite definir y desplegar un menú de opciones permitiendo la selección de uno de ellos.
 - xi. MnuBtn, permite definir y desplegar un menú de botones permitiendo la selección de uno de ellos.
 - xii. MnuBar, permite definir y desplegar un menú de barra (hasta 9 opciones con 9 sub opciones cada una).
 - xiii. MnuNav, permite definir y desplegar un menú de lista con estructura jerárquica (con hasta 18 opciones).
- c. Cada formato predefinido utiliza específicas variables, que por lo general están agrupadas por afinidad de información. Una coordinada secuencia y distribución de los mismos, en la pantalla, permite una interfaz muy aceptable y controlada.
- d. En cada requerimiento de formato se pueden aplicar los controles necesarios, minimizando de esta manera la concurrencia de errores y la interrelación de los controles.

12. Por lo general en toda sistematización se requiere la paginación de los registros o filas de una tabla, como se resuelve esta necesidad en ABG.
- a. Para esta necesidad ABG tiene previsto la utilización de la sentencia SflExc, la cual requiere una serie de estructuras de códigos, algunas obligatorias y otras optativas, para resolver la paginación de determinadas columnas de una o más tablas relacionadas.
 - b. Todas las estructuras necesarias deben codificarse en un misma Transacción/Evento o programa ABG, al cual se accede mediante la especificación de la sentencia SflCall, a través de la cual se especifica la identidad Transacción/Evento o Programa ABG y opcionalmente los parámetros que sean necesarios.
 - c. Estructura obligatoria para efectuar la definición del Sub File, en la misma corresponde especificar:
 - i. SflHdr, necesaria para establecer el título de cabecera para los registros o filas paginadas, es posible tener una cabecera de hasta 4 líneas.
 - ii. SflKey, utilizadas para especificar la lista de los campos necesarios para identificar a los registros paginados, con el objeto de recuperarlos para:
 1. Retornar la identidad.
 2. Aplicarle una acción de las permitidas.
 - iii. SflOpc, permite definir una lista (descriptivas) de las acciones permitidas.
 - d. Estructura obligatoria para definir la sentencia SqlSlt de ABG, la cual permite especificar las columnas que se recuperarán de una o más tablas y el ordenamiento de las mismas.
 - e. Estructura obligatoria para definir, mediante la sentencia SflCol de ABG, a cada una de las columnas que se desean visualizar en la paginación.
 - f. Estructura opcional para definir la codificación necesaria para procesar a cada registro recuperado por la consulta, esto posibilita la formación de determinadas variables a partir de algunas de las recuperadas.
 - g. Estructura opcional para el procesamiento de cada registro seleccionado en la paginación, con el objeto de aplicarle una acción.
 - h. Estructura opcional para la construcción de un filtro adicional, esto consiste en la posibilidad de requerir datos para construir condiciones que se adicione a la cláusula WHERE de la estructura de selección.
13. Basándonos en lo anteriormente expresado, observamos que solamente se hacen definiciones de código, y que en ningún momento se especifica una lógica para el funcionamiento del Sub File, lo que hace suponer la existencia de una lógica en la operatividad del Sub File, cual es?
- a. La sentencia SflExc se encarga de incorporar la lógica para la operatividad del Sub File, la cual prevé la siguiente funcionalidad:
 - i. Desplazamiento horizontal, en las columnas visualizadas, hacia ambos lados (izquierda y derecha).
 - ii. De acuerdo al formato utilizado se determina una cantidad de registros que se visualizaran por cada página, por lo tanto se habilita un desplazamiento hacia delante y atrás de los registros o filas paginadas.
 - iii. Activación de la estructura de filtro adicional.
 - iv. Selección de un registro o fila con el objetivo de aplicarle una de las acciones previstas en el código de la estructura de definición.
 - v. Listado de los registros visualizados.

14. Para construir una Transacción/Evento o Programa ABG, que realice un listado contemplando totalizaciones grupales (cortes de control). Cual es el código que se necesita especificar.
- a. Todas las estructuras necesarias para producir un reporte deben codificarse en un misma Transacción/Evento o programa ABG, al cual se accede (desde el evento invocador) mediante la especificación de la sentencia Call, especificando en la misma la identidad Transacción/Evento o Programa ABG y opcionalmente los parámetros que sean necesarios.
 - b. Corresponde definir, a través de la sentencia DefVar, a todas las variables que sean necesarias, siempre y cuando las mismas no estén consideradas como globales.
 - c. Mediante la utilización del asistente de diseño corresponde definir las distintas sesiones que constituyen a los reportes, el orden y la denominación de las sesiones son:
 - i. De cabecera de grupo (HDR-CC0 a HDR-CC1 a HDR-CC9).
 - ii. De detalle (DET-DET).
 - iii. De totales de grupo (FOO-CC9 a FOO-CC1 y FOO-CC0).
 - d. Es necesario utilizar la sentencia Exec de ABG, para ejecutar un comando del sistema operativo y que en este caso en particular corresponde ejecutar el comando OVRPRTF para asignar los atributos de la salida de impresión a generar.
 - e. A continuación corresponde ejecutar la sentencia Report de ABG, la cual requiere referenciar varias estructuras de códigos, algunas obligatorias y otras optativas, para generar el reporte correspondiente. Además debe especificarse el nombre de la etiqueta que contiene al diseño del reporte, a partir de las distintas sesiones definidas en el mismo, se infiere la lógica necesaria que debe ser incluida en el programa ABG. Se pueden definir hasta 9 cortes de control, más un corte general y para cada corte se pueden especificar cabecera y pie.
 - f. Estructura obligatoria para definir la sentencia SqlSlt de ABG, a través de la cual corresponde especificar las columnas que se recuperarán de una o más tablas y el ordenamiento de las mismas, es necesario destacar que este último debe corresponder con las sesiones que se definan.
 - g. Estructura opcional para definir la codificación necesaria para procesar a cada registro recuperado por la consulta, esto posibilita la formación de determinadas variables a partir de algunas de las recuperadas.
 - h. La lógica que se genera para la Transacción/Evento o Programa ABG, prevé la invocación a determinadas estructuras que pueden codificarse en el caso de ser necesarias, con el objetivo de satisfacer determinadas particularidades. Estas estructuras referidas son:
 - i. CHD-CC1 a CHD-CC9 o CHD-CC0 invocada cuando se procesa el primer registro del grupo de control correspondiente, esto posibilita codificaciones necesarias para la impresión de la sesión asociada (ejemplo, acceder a una tabla relacionada una única vez, evitando realizar la relación en el SQL de selección, permitiendo optimizar la ejecución del proceso).
 - ii. AHD-CC1 a AHD-CC8 o AHD-CC0, invocada después de imprimir la sesión de impresión asociada.
 - iii. AFT-DET, invocada después de imprimir la sesión de impresión de detalle (DET-DET).
 - iv. CFO-CC1 a CFO-CC9 o CFO-CC0 invocada después de procesar el último registro del grupo de control correspondiente, esto posibilita codificaciones necesarias para la impresión de la sesión asociada.
 - v. AFO-CC1 a AFO-CC1 o AFO-CC0 invocada después de imprimir la sesión de impresión asociada.

- vi. CFO-OVF, invocada cuando se detecta el fin de página antes de invocar la impresión de la sesión establecida como overflow, esto posibilita la codificación necesaria en esta instancia del reporte.
 - vii. PHD-OVF, invocada después de imprimir la sesión establecida como overflow, esto posibilita la codificación necesaria en esta instancia del reporte.
15. Para construir una Transacción/Evento o Programa ABG, para que realice una exportación de información, cual es el código que se necesita especificar.
- a. Todas las estructuras necesarias para producir una exportación de datos deben codificarse en un misma Transacción/Evento o programa ABG, al cual se accede (desde el evento invocador) mediante la especificación de la sentencia Call, a través de la cual se especifica la identidad Transacción/Evento o Programa ABG y opcionalmente los parámetros que sean necesarios.
 - b. Corresponde definir, a través de la sentencia DefVar, a todas las variables que sean necesarias, siempre y cuando las mismas no estén consideradas como globales.
 - c. A continuación corresponde ejecutar la sentencia GenTxt de ABG, la cual requiere una serie de estructuras de códigos, algunas obligatorias y otras optativas, para generar la exportación correspondiente.
 - d. Estructura obligatoria para definir la sentencia SqlSit de ABG, a través de la cual se especifican las columnas que se recuperarán de una o más tablas y el ordenamiento de las mismas. Esto constituye el conjunto base para la exportación.
 - e. Estructura obligatoria para definir, mediante la sentencia ColTxt de ABG, a cada una de las columnas que se desean exportar.
 - f. Estructura condicional (solo cuando la exportación es tipo *XML) para definir, mediante la sentencia ColTxt de ABG, a cada una de las cabeceras de las columnas que se desean exportar.
 - g. Estructura opcional para definir la codificación necesaria para procesar a cada registro recuperado por la consulta, esto posibilita la formación de determinadas variables a partir de algunas de las recuperadas.
16. La programación ABG, soporta la necesidad de procesamiento a través de internet?.
- a. Totalmente y de manera sencilla, puesto que se utilizan:
 - i. Un programa CGI para establecer la conexión, se encarga de asignarle una identidad a cada cliente que se conecta y de devolverle el formulario de logeo con enlace al programa de control de usuario.
 - ii. Un segundo programa que se encarga de realizar el control de los usuarios que intenta conectarse. Las funciones realizadas son:
 - 1. Controlar el número de conexión asignado, contemplando el estado de la misma y el tiempo de inactividad, si el primer control es satisfactorio se procede a controlar la identidad y contraseña.
 - 2. Si los controles no fueron satisfactorios se devuelve un formulario con indicación del error y con enlace al primer programa CGI.
 - 3. Si los controles fueron satisfactorios, se procede a enlazar al tercer programa CGI con especificación de la Transacción/Evento que contiene al menú principal de la operatividad.
 - iii. Un tercer programa CGI que se encarga de invocar al Transaccionador ABG con la identidad de la Transacción/Evento enlazada a través del formulario, para gestionar los distintos requerimientos.

- iv. Cada Transacción/Evento funciona de la siguiente manera:
 - 1. Recuperar los valores de las variables CGI, esto se realiza mediante la sentencia VarCgi de ABG.
 - 2. Efectuar el procesamiento necesario, generando mediante la sentencia FrmHtml de ABG un formulario o con la sentencia Report producir un reporte con salida HTML.
- b. La solución se implementa mediante programas CGI, para lo cual se requiere:
 - i. Configuración del Servidor Apache, incluido en el Sistema Operativo OS400.
 - ii. Activación del Servicio HTML.
 - iii. Activación del Servicio ABGWEB.
 - iv. Diseñar los formularios HTML, los cuales pueden contener sentencias Scrip (Java o Visual Basic). Estos formularios deben ser exportados por el entorno de programación de ABG y los mismos son almacenados por el diseñador de formularios.