

TABLA DE CONTENIDO

Capítulo IV Definiciones De Nómina.....	3
Ficheros.....	3
Configuración de Empresa	3
Empresa	3
Antigüedad E Interés.....	5
Vacaciones.....	6
Préstamos	7
Otros	7
Cuentas Contable e I.T.F.	8
Clasificación De Conceptos.....	9
Conceptos De Pago	9
Tipos De Conceptos.....	11
Identificación	12
Componentes para el Diseño del Concepto.....	15
Uso de los Valores Desde Otras Tablas	16
Uso De Variables	17
Configuración	22
Comentarios.....	24
Otras Consideraciones	24
Utilización De Comentarios y Campos Memos o Textos	26
Tablas Temporales Para Prenómina.....	27
Configuración Del Concepto	28
Calculo De Salarios Promedios.....	28
Descripción De Algunos Conceptos Incluidos Como Ejemplos.....	30
Constantes	30
Jornadas y Turnos.....	30
Días Feriados	31
Otras Nóminas.....	31
Cuentas Contables	32
Tipos De Ausencias.....	32
Tasas De Intereses	30
Carta Para El Trabajador.....	32

Unidad Tributaria	32
Tarifa De I.S.L.R. (Art. 50)	33
Fechas y Calendario	33
Históricos	41
Tablas Asociadas	45
Ficha Del Trabajador	49
Salarios	50
Intereses	52
Funciones Xbase	54
Otras Funciones	58
Informe	61
Listados De Conceptos De Pagos	62
Formulas De Los Conceptos	62
Constante	62
Tasa De Interés	62
Proceso	63
Importar Ejemplo	63
Importar Datos Desde Dpnm24	63
Exportar Conceptos	64
Importar Conceptos	64
Otros	64
Sistema Administrativo Contable	64
Ficha Del Trabajador	65
Denominación De Términos	65

CAPÍTULO IV DEFINICIONES DE NÓMINA

Autonomía e Inmediatez son los principales beneficios que lograrás con ADAPTAPRO NOMINA.

Definiciones de Nómina: Provee las funcionalidades que te permiten indicarle al sistema las instrucciones y parámetros que generan los resultados de nómina.

Los conceptos de pago estos compuestos por instrucciones contentivas de los procedimientos y cálculos que generan los resultados y están escritos en el léxico DpXbase facilitando superar los límites de las antiguas ecuaciones lineales de sus antecesores “ADAPTAPRO NOMINA”

Esta aplicación ubicada como tercera opción dentro del menú de aplicaciones en el Sistema de ADAPTAPRO NOMINA, contenido de los componentes específicos para la definición de los cálculos que genera los resultados para generación de los recibos de pago.

FICHEROS

Agrupar todas las tablas maestras de este módulo, se especificará la creación de dichas tablas y las mismas serán llamadas o utilizadas en otros módulos. Su contenido es introducido por el usuario y no deriva de algún proceso.

CONFIGURACIÓN DE LA EMPRESA

Este formulario permite definir los datos de la empresa, el uso de los conceptos y las diferentes formas de aplicar los conceptos de pago. Las definiciones son de carácter general por cada empresa y están totalmente vinculadas con la programación de conceptos, Reportes y Consultas. Por tal motivo los resultados pueden variar en diferentes configuraciones razón necesaria para que el usuario defina los criterios de Nómina antes de realizar los procesos de actualización de la primera nómina.

El programa presenta una ventana de diálogo segmentada en varias carpetas diseñadas específicamente para cada componente de definición:

EMPRESA

Solicita los datos formales de la empresa necesarios para la emisión de Reportes y Cálculos de las distintas planillas, entre los valores están:

Dirección: Está compuesto por tres líneas de cincuenta (50) caracteres cada una y registra la dirección de la empresa. La dirección está representada por los valores: oDp:cDir1, oDp:cDir2 y oDp:cDir3

Teléfonos: Está compuesto por tres líneas de catorce (14) caracteres cada una y registra los números telefónicos del Patrono y están representados por los valores: oDp:cTel1, oDp:cTel2 y oDp:cTel3

R.I.F.: Acepta veinte (20) caracteres para registrar el número del “R.I.F.” del patrono. El sistema no aplica ningún tipo de verificación.

Ministerio del Trabajo: Acepta catorce (14) caracteres para registrar el número de inscripción o registro en el Ministerio del Trabajo por parte del patrono, el sistema no realiza ningún tipo de validación.

Seguro Social: Acepta catorce (14) caracteres para registrar el número de Inscripción del Patrono en el “I.V.S.S.” y sin validación por parte del Sistema.

N.I.T.: Acepta catorce (14) caracteres para registrar el número del “N.I.T.” del patrono. El sistema no aplica ningún tipo de verificación.

Utilizar Fechas en Históricos: se refiere al campo de la fecha que será utilizada para establecer los rangos de búsqueda para la tabla histórico solicitados por las funciones: ACUMC_FCH() , ACUMV_FCH() y demás funciones referidas con fechas, éste campo presenta tres alternativas: Desde, Hasta y Sistema, Cabe destacar que los resultados obtenidos desde la tabla histórica pueden variar según el tipo de fecha utilizada, como sucede clásicamente en el tipo de nómina semanal y se solicita el acumulado de un determinado concepto desde el primero del mes hasta el último día del mes, si es tomando en cuenta la fecha “Desde” genera un resultado diferente que si se toma la fecha “Hasta”, igualmente genera otro resultado si se toma la fecha del sistema. Las fechas corresponden a la tabla de recibos y estas se encuentran relacionadas con la tabla histórica de pagos, la fecha “Desde” corresponde al inicio del periodo de pago, la fecha “Hasta” corresponde al final del periodo de pago y la fecha del “Sistema” corresponde a la fecha de procesamiento o generación del resultados según la fecha del sistema representada por “oDp:dFecha”.

Inicio de la Semana: Presenta siete alternativas: Lunes, Martes, Miércoles, Jueves, Viernes, Sábado y Domingo, se refiere al día de la semana en que se inicia el periodo de pago “Semanal” necesario para que el sistema determine el periodo de pago representado por las variables “oDp:dDesde” y “oDp:dHasta”.

Forma de contabilizar: Presenta cuatro alternativas para contabilizar por: concepto, departamento, unidad funcional y grupo. Se refiere a la forma de realizar la contabilidad de la nómina.

Distribución Monetaria: Presenta una línea de sesenta y tres (63) caracteres con la finalidad de aceptar en forma ascendente y separados por comas “,” las diferentes monedas para realizar los pagos en “Efectivo o Metálico” que serán reflejados en los recibos de pago. Está

representado por "oDp:cBilletes" y es utilizado por los programas "Prenómina y Nómina" para crear una tabla asociada al trabajador que contiene el desglose monetario además genera el total de monedas requeridas según los recibos de pago para la forma de pago "Efectivo". Cabe destacar que los formatos de recibos y prenóminas diseñadas con Crystal Report están adaptados para Quince (15) monedas. En caso de ser necesario emplear más o menos monedas el sistema ajusta automáticamente las tablas que contienen la distribución monetaria y es necesario que el usuario redefina los formularios de impresión a través de Crystal Report.

Redondeo de 5 a 0: Activa el proceso de redondeo o aproximación decimal generado por el proceso de ejecución de conceptos de pago para Prenómina y Actualización de Nómina. Ejemplo: Si la fracción decimal del resultado es 1.45 será igualado a 1.00, si es 1.57 será 1.50.

Detectar Carnaval y Semana Santa: Al activar esta opción, el proceso de cálculo de Prenómina, Vacaciones y Nómina que utilizan funciones de fechas vinculadas con el calendario detectan e incluyen dos días correspondientes a lunes y martes de Carnaval del año en curso y de la misma manera el jueves y viernes de Semana Santa. Éste evita declarar anualmente los días de Carnaval y Semana Santa en la tabla de calendario de días feriados, debido a que cada año corresponde a fechas diferentes. Si la opción esta inactiva, debe declararlos en la tabla de días feriados.

ANTIGÜEDAD E INTERÉS

Representa a un conjunto de parámetros vinculados directamente con los conceptos y definiciones para la Antigüedad Laboral e Intereses.

Antigüedad Laboral: Solicita el código del concepto de pago que representa la antigüedad laboral artículo 142 necesarios para los formularios de consulta de antigüedad laboral, importación de datos, generación de antigüedad y cálculo de intereses.

Anticipos de Antigüedad: Solicita el código del concepto de pago que representa el monto por anticipos de antigüedad laboral requeridos por el cálculo de intereses sobre antigüedad laboral y los estados de cuenta por el mismo motivo.

Concepto Pago de Intereses Sobre Antigüedad: Solicita el código del concepto de pago de Intereses sobre Antigüedad Laboral necesarios para el cálculo de intereses sobre antigüedad laboral y los estados de cuentas por el mismo motivo.

Inicio para el Cálculo de Intereses: Solicita la fecha para que el sistema inicie el cálculo de intereses por concepto de antigüedad laboral. En algunos casos la empresa ya ha calculado y/o pagado los intereses sobre antigüedad laboral antes de iniciar el uso de ADAPTAPRO NOMINA. La información de los intereses calculados y/o pagados pueden ser cargados mediante la nómina de tipo "DI" Datos Iniciales o a través de la opción: "Importar antigüedad e

Intereses". Ambos procesos generan los valores para el concepto "A411 Pago de Intereses sobre Antigüedad"

Indizar Intereses sobre Antigüedad: Activa el proceso de capitalización de intereses obtenidos del cálculo de intereses mediante la función "INTERES()" aplicado a los montos históricos asociados con los conceptos: "H400 Compensación Mensual Antigüedad Art. 142", "A073 Anticipo de Antigüedad" y "A411 Pago de Intereses sobre Antigüedad"

Distribuir Días Adicionales Art. 142: Permite distribuir los días adicionales entre los doce meses del año.

Incluir en Nómina 2 Días Adicionales Art. 142: Permite tomar la decisión si se desea o no se desea incluir o pagar los 2 Días Adicionales en la nómina normal (semanal, quincenal o mensual), el sistema toma como fecha para el pago la fecha de ingreso del trabajador.

Pagar Intereses en Fecha Aniversario: Permite pagar o cancelar los intereses en la fecha aniversario del trabajador en la nómina normal (semanal, quincenal o mensual).

Incluir Artículo 104 en Liquidación: Permite incluir o pagar el Artículo 104 al momento de generar la Liquidación.

Diferir Pagos de Utilidades para Liquidación: Permite posponer el pago de utilidades para el inicio del siguiente año. Aplicaría para aquellas empresas que esperan finalizar el ejercicio económico para pagar las utilidades.

VACACIONES

Representa a un conjunto de parámetros vinculados directamente con los conceptos y definiciones para las vacaciones y bono vacacional del trabajador.

Días de Vacaciones: Solicita el código del concepto de pago que representan los días de vacaciones a pagar para los formularios de consultas y generación de vacaciones.

Reintegro de días de Vacaciones: Solicita el código del concepto de pago que representan los días acumulados de vacaciones para los formularios de consultas y generación de vacaciones.

Almacenar Próximo Disfrute: Solicita el código del concepto que almacena el próximo disfrute de vacaciones. Para que el sistema pueda realizar las consultas y la generación de las vacaciones.

Días Adicionales de Disfrute: Solicita el código del concepto que almacena los días adicionales de disfrute. Para que el sistema pueda realizar las consultas y la generación de las vacaciones.

Días Disfrutados: Solicita el código del concepto que almacena los días de vacaciones disfrutados. Para que el sistema pueda realizar las consultas y la generación de las vacaciones.

Vacaciones: Permite seleccionar si la empresa paga vacaciones colectivas o individuales.

Bono Vacacional: Permite seleccionar si la empresa paga el bono vacacional en las vacaciones ó cuando el trabajador cumple años en la empresa.

Periodo Vacacional: Permite colocar la fecha de inicio y fin de las vacaciones para las vacaciones colectivas.

Nota: Sin embargo, el sistema permite el manejo de las vacaciones colectivas e individuales al mismo tiempo. Para aquellas empresas que manejen los dos tipos de vacaciones se recomienda generar primero las vacaciones individuales y luego las vacaciones colectivas.

PRÉSTAMOS

Representa a un conjunto de parámetros vinculados directamente con los conceptos y definiciones para el manejo de préstamos e intereses sobre préstamos.

Otorgamiento de préstamos: Solicita el código del concepto de pago que representa el otorgamiento del préstamo o la asignación del mismo, necesarios para los formularios de consulta y generación de préstamos.

Deducción de préstamos: Solicita el código del concepto de pago que representa la deducción de préstamos o la cancelación del mismo, necesarios para los formularios de consulta y generación de préstamos.

Intereses sobre préstamos: Solicita el código del concepto de pago que representa los intereses sobre préstamos, necesarios para los formularios de consulta y generación de préstamos.

Aplicar Intereses sobre préstamos: Permite seleccionar si la empresa aplica intereses sobre préstamos.

OTROS

Solicita la configuración de Otros conceptos y el uso general para el procesamiento de la nómina.

Consulta de Otros Conceptos desde DpNmWeb: Activa o Inactiva la posibilidad de que el trabajador pueda consultar otros conceptos “Distintos a Asignaciones y Deducciones” desde la aplicación DpNmWeb.

Diferir Retención del S.S.O. para fin de mes: Activa o Inactiva el diferimiento de retención del Seguro Social de la primera quincena de los trabajadores asociados con la nómina “Quincenal”. El monto de la retención es almacenado en el Concepto “H104 S.S.O. Diferido” y será sumado a la deducción del Concepto “D004 Retención S.S.O.” cuando se ejecute la nómina de la segunda quincena.

Diferir Retención de Régimen Prestacional de Empleo para fin de mes: Activa o Inactiva el diferimiento de retención del “Régimen Prestacional de Empleo” de la primera quincena de los trabajadores asociados con nómina “Quincenal”. El monto de la retención es almacenado en el Concepto “H105 Régimen Prestacional de Empleo” y será sumado a la deducción del Concepto “D005 Retención Régimen prestacional de Empleo” cuando se ejecute la nómina de la segunda quincena.

Diferir Retención de L.V.H. para fin de mes: Activa o Inactiva el diferimiento de retención del “L.V.H.” de la primera quincena de los trabajadores asociados con nómina “Quincenal”. El

monto de la retención es almacenado en el Concepto “H128 L.V.H. Diferido” y será sumado a la deducción del Concepto “D028 Retención Ley de Vivienda y Hábitat” cuando se ejecute la nómina de la segunda quincena.

Consideraciones:

Luego de ejecutar la primera nómina, no cambie los valores de la configuración ya que puede causarle cálculos diferentes a los que inicialmente fueron planificados.

CUENTAS CONTABLES I.T.F.

% del Impuesto a las transacciones Financieras: Permite colocar el porcentaje o la tasa del (ITF), la misma es utilizada en el proceso de contabilización administrativo y contabilidad.

Cuentas Contables I.T.F.: Permite colocar la cuenta contable del ITF, la misma es utilizada en el proceso de contabilización administrativo y contabilidad.

Cuentas Contables Nómina por Pagar: Permite colocar la cuenta contable de nómina por pagar, la misma es utilizada en el proceso de contabilización administrativo y contabilidad.

CLASIFICACIÓN DE CONCEPTOS DE NÓMINA

Esta opción consiste en una tabla que permite definir las diferentes agrupaciones o clasificaciones de los conceptos de pago, cada clasificación se identifica a través de un código de veinticinco (25) caracteres y una descripción de cuarenta y cinco (45) caracteres.

La clasificación facilita la agrupación de uno o varios conceptos de pago bajo una identificación que permita asociarlos con usos específicos, ejemplo: utilidades, vacaciones, liquidaciones, etc. Es decir, facilita la administración de los conceptos de pago, aunque su clasificación no altera ningún tipo de resultado o proceso, sin embargo, si permite extraer "por Clasificación" información desde la tabla histórica de conceptos de pagos a través de las funciones: ACUMULA_CLA () y AVERAGE_CLA ()

Cabe destacar que es posible obtener valores promedios relativos o average según los pagos relacionados con conceptos asociados a una determinada clasificación. Ejemplo: Se requiere emplear varios salarios promedios o exclusivos para: "Utilidades", "Vacaciones" y "Prestaciones Sociales", para lograrlo es necesario crear una clasificación por cada requerimiento y relacionar todos los conceptos vinculados con cada promedio requerido, luego el empleo de la función AVERAGE_CLA (dDesde,dHasta,cCodCla) determinará el valor promedio.

Clasificar Conceptos:

El proceso de asociación de conceptos con clasificaciones puede realizarse de dos formas:

1. Por medio del **Catálogo de Conceptos de Pago** disponible en: Aplicaciones => Definiciones de Nómina => Ficheros => "Conceptos de Pago", haciendo clic en el icono



(Clasificar Conceptos).

2. Ó también desde el **Catálogo de Clasificación de Conceptos** a través del botón  (Clasificar Conceptos) cuya ubicación en el sistema es: Aplicaciones => Definiciones de Nómina => Ficheros => "Clasificación de Conceptos de Nómina".

El botón  presenta una lista que contiene todos los conceptos de pagos indicando: Código, descripción y un checkmark para asociar el concepto (Concepto no asociado , concepto asociado). Para activar o inactivar la asociación sólo es necesario hacer Clic en este último y automáticamente se actualiza la tabla de asociación entre conceptos y clasificación.

CONCEPTOS DE PAGO

Permite la inclusión o modificación de los conceptos de pago que conforman los rubros del recibo de pago. Cada concepto se refiere a una fórmula o programa que genera resultados para la nómina, se identifica por medio de un código "Xnnn" de cuatro (4) caracteres donde el primero "X" indica el tipo del resultado: **A**signación, **D**educción, **N**eutros, **F**unciones é **H**istóricos y los tres restantes "nnn", indican el orden de ejecución: 001, 002, 003, 004, 005. La nomenclatura de ambos conforma el código del concepto.

La fórmula debe ser escrita en forma procedural, es decir, un programa conformado por instrucciones que se inicia con la declaración de una función, seguido por una serie de cálculos y que debe retornar un valor como resultado. Este nuevo modelo de formulación supera los productos anteriores donde se programaba una fórmula de manera lineal con limitaciones; es decir, un concepto en una línea de instrucción; ahora, ADAPTAPRO NOMINA incorpora una nueva forma que es prácticamente un programa contentivo de: instrucciones y sentencias "IF/ELSE/ENDIF FOR/NEXT WHILE/ENDDO CASE/ENDCASE", acepta parámetros y devuelve valores, además; puede contener sus propias funciones y variables, permitiendo así posibilidades infinitas para lograr cualquier cálculo de pago.

El concepto Open Source de AdaptaPro, dispone los programas fuentes en el repositorio [GITHUB](#)

1. Todos los conceptos de pago en archivo [nmconceptos.zip](#)
2. Todos los programas [fuentes de cada concepto de pago](#).
3. Programa fuente de la clase [TNOMINA](#) contentiva de la ejecución de la nómina.
4. Programa fuente de las [funciones de AdaptaPro Nómina](#).
5. Programas fuentes escrito en [DpXbase del módulo de nómina](#).
6. [Capítulo1](#) Instalación e implementación.
7. [Capítulo2](#) Ejecución de Nómina.
8. [Capítulo3](#) Recursos Humanos.
9. [Capítulo4](#) Definiciones de Nómina.
10. [Capítulo5](#) Administración del Sistema
11. [Capítulo6](#) Programación DpXbase
12. [Generador de Reportes](#).

Optimización en el cálculo de nómina

Para reducir el tiempo de respuesta, es necesario utilizar sólo los conceptos que necesita, "Inactive los conceptos que no sean requeridos", logrando que el sistema efectúe la menor cantidad de cálculos por ecuación. Igualmente debe evitar el uso innecesario de la variable "nFactor1, nFactor4", debido a que estos valores son almacenados en una tabla asociada (dicha tabla es interna del programa y guarda las variables que pueden ser utilizadas públicamente), empléelo únicamente cuando desee almacenar cualquier valor obtenido en la ejecución de las fórmulas. Ejemplo: En algunos casos es necesario almacenar la cantidad de días para las vacaciones o el salario empleado para las utilidades, para lograrlo es necesario utilizar cualquier

variable para los factores, "nFactor1: =SALARIO". nFactor1 es almacenado en la tabla de observaciones y factores.

Los contenidos de las fórmulas no forman parte del sistema, los resultados que generan son obtenidos por la ejecución de los conceptos.

TIPOS DE CONCEPTOS

1. **Asignaciones:** Su primer carácter es "A" y todos los resultados deben ser de signo positivo, ejemplo: A001 (Salario Semanal) o A002 (Salario Quincenal) etc.
2. **Deducciones:** Se inicia con la letra "D" y debe ser el único tipo de concepto para generar resultados de signo negativo. Ejemplo: D001 (Inasistencia) D002 (Retardos en Horas).
3. **Neutros:** Se inician con la letra "N", su único objetivo es procesar resultados sin alterar las asignaciones y deducciones. Ejemplo: N001 (Cuenta para Pagar Otro Préstamo), etc. También son útiles para solicitar variaciones que puedan afectar los cálculos.
4. **Funciones:** Se inician con la letra "F", no generan resultados de manera automática, solo cuando son invocados desde cualquier concepto a través de la función "FUNCION (nnn)". Ofrece una gran utilidad gracias a que un concepto de tipo función trabaja como servidor de resultados para cualquier otro concepto, simplificando fórmulas y minimizando procesos. Ejemplo: En muchos conceptos es necesario utilizar el valor de la hora de trabajo. Si el concepto [F001](#), contiene esta fórmula, todos los demás conceptos que requieren este resultado solo deben invocarlo mediante "FUNCION (001)", y de esta forma se evita la repetición de dicha fórmula en cada uno de los conceptos.
5. **Históricos:** Se inician con la letra "H" y no afectan ni generan ningún tipo de pago, tienen como finalidad acumular valores en la tabla histórica de pago, posteriormente pueden ser utilizados en otros cálculos, ejemplo: El concepto ["H400"](#) que calcula y almacena mensualmente el valor de la "Antigüedad Laboral", luego a través del concepto "A401" se realiza el pago por "Antigüedad Laboral" obtenido de los valores acumulados de "H400". Igualmente, el cálculo de interés puede ser aplicado sobre el acumulado de cualquier concepto.
6. **Presupuestos:** Se inician con la letra "P", utilizado para Otra Nómina "Presupuesto Anual"
7. **Utilización de Otras Letras Iniciales:** Es posible implementar el uso de otra letra para definir tipos de nómina según criterio del usuario. Ejemplo de ello está en la implementación de la nómina presupuestaria anual, quien utiliza los conceptos cuyos códigos se inician con la letra P, dentro su programa incluye la condición: IF

CONDICION="A" .AND. oNm:cOtraNom="PA"

Ejemplo: el concepto Xnnn puede utilizar cualquier variable generada por un concepto ejecutado anteriormente, D004, puede utilizar las variables: C_A000...C_A999, C_B000...C_B999, C_C000...C_C999, C_D000...C_D003.

Los conceptos de pago están compuestos por cuatro secciones: Identificación, Cálculo, Configuración y Comentarios.

Ejecución secuencial: La ejecución de los conceptos solo se realiza cuando se requieren resultados para la nómina, mediante el acceso secuencial de los datos de cada uno de los trabajadores. Por cada trabajador se ejecutan todos los conceptos organizados de la siguiente manera: primero: "A" (asignaciones), luego, "B","C", "D" (deducciones), hasta concluir con la "Z", omitiendo la "F" (funciones) y según su numeración, ejemplo: "A000", "A001"... hasta "A999". Cada concepto ejecutado genera una variable residente que contiene el valor resultante de la fórmula y ésta es disponible solo por cada trabajador accedido en los procesos, esta variable se denomina : C_xnn, permite que otros conceptos puedan interactuar entre sus resultados según el orden de precedencia, Ejemplo: Al ejecutar los conceptos "A001" y "A002", primero se genera la variable "C_A001" y el concepto "A002" puede utilizar el resultado del concepto "A001" utilizando la variable "C_A001", pero A001, no puede utilizar el valor de "C_002" porque esta se genera después de "C_A001", igualmente para los demás conceptos.

IDENTIFICACIÓN

- **Código:** Acepta hasta cuatro (4) caracteres, donde el primero debe ser una letra y los tres restantes números que conforman el código de identificación del concepto "Xnnn". La interpretación que el sistema le dará al concepto dependerá de esta letra inicial, así para: **A**signación (Annn), **D**educción (Dnnn), **N**eutros (Nnnn) o **F**unción (Fnnn). Las demás letras quedan disponibles para la utilización que el usuario requiera, solo que no aparecerán en los informes de pre Nómina ni recibos de pago. Los conceptos "Dnnn" son los que generan resultados con valor negativo.

- **Descripción:** Acepta cuarenta (40) caracteres y se refiere a la descripción del concepto que será mostrado en los recibos de pago.

- **Activo** Activa o Inactiva el concepto, si es inactivo será excluido de todos los procesos de cálculo sin necesidad de eliminar el registro; además, el sistema detectará la

cantidad de fórmulas que utiliza este concepto para impedir la desactivación en caso que la búsqueda sea exitosa.

CÁLCULO

Cálculo

Provee un editor de textos de tipo que permite escribir las condiciones y fórmulas que generan el resultado del concepto. Esta fórmula se escribe bajo el léxico Xbase procedualmente, compuesto por funciones e instrucciones que se ejecutan de manera ascendente. Prácticamente un concepto contiene un programa bajo la siguiente regla:

1. Se inicia con la declaración de una función, según:

FUNCTION A001 (uPar1, uPar2...uPar4), El sistema sugiere que se denomine igual que el código del concepto, dentro de ella se declaran los posibles parámetros "separados por comas" que pueden recibir en el momento de ser ejecutada. Ver función A001 (uPar1, uPar2...uPar4)

2. Declaración de variables locales "Sólo tienen ámbito dentro de la función", ejemplo: LOCAL nValor:=0,dFecha:=CTOD(""),cChar:=SPACE(10),lLogico:=.T.

Estas variables son declaradas e iniciadas con sus respectivos valores, no existe nomenclatura en la definición de variables según el tipo (nValor:=0,dFecha:=CTOD(""),cChar:=SPACE(10),lLogico:=.T.), es recomendable colocar como prefijo EN MINÚSCULA el tipo de dato que soporta, así se logra comprender fácilmente el programa.

Importante: *Sólo declare las variables que necesite.*

3. Condición para calcular, define una bifurcación en el programa que inicia con la cláusula IF <Condición>, el resultante verdadero concluye en la cláusula ELSE y la bifurcación resultante falsa finaliza con la cláusula ENDIF

Ejemplo:

```
IF CONDICION="A" .AND. FECHA_ING>oNm:dDesde
    nResult:=SALARIO*DIAS_HAB()
ELSE
    nResult:=SALARIO
ENDIF
```

4. **Cálculos:** Se derivan de la conjunción de diversos elementos: Variables, campos, funciones, valores de las tablas, operadores de tipo "lógicos, aritméticos y relacionales". Ejemplo: nResult:=DIAS_HAB()*SALARIO

Emplea como contenedor del resultado a la variable "nResult", asigna el valor a través del operador "=", utiliza la función DIAS_HAB() para determinar la cantidad de días hábiles en el periodo de pago, lo multiplica "*" por el valor del Campo SALARIO obtenido desde la ficha del trabajador.

Así mismo el cálculo final puede derivarse de la conjunción sistemática de una expresión compuesta por diversos elementos, tales como: Variables, Campos y/o Funciones.

5. Retorno del resultado, es realizado por la cláusula RETURN nResult, además indica la finalización en la ejecución del programa "Concepto".

6. Empleo de funciones definidas por el usuario: Pueden ser consideradas como sub-programas que pueden ser invocados desde el concepto y contribuyen a simplificar el

desarrollo de un concepto, el nombre no puede superar diez (10) caracteres y tampoco puede coincidir con el nombre de una función ya existente dentro del sistema de nómina.

Ejemplo:

```
FUNCTION A001 (nPar1, nPar2)
  IF CONDICION="A"
    nResult:=DIALUNES ()
  ENDIF
  RETURN nResult

FUNCTION DIALUNES ()
  LOCAL nLunes
  LOCAL dDesde:=oNm: dDesde

  WHILE dDesde<=oNm:dDesde
    IF DOW(dDesde)=2 // Lunes
      nLunes++
    ENDIF
    dDesde:=dDesde+1
  ENDDO
  RETURN nLunes
```

Utilice en símbolo “;” para continuar en la siguiente línea el contenido de la instrucción o sentencia, ejemplo:

```
IF CONDICION="A" .AND.;  
FORMA_PAG="E" .AND.;  
SALARIO>=1000
```

7. Para que el concepto cumpla las condiciones exigidas por el formulario de Variaciones, debe indicar “lVariac:=.t.” dentro de la fórmula, que le indica al programa de variaciones que el concepto califica o cumple la condición para solicitar variaciones. También debe tomar los requerimientos de su configuración para ser aceptado por el formulario de variaciones, cabe destacar que el concepto debe cumplir dos condiciones, la primera es estática, se define en la sección de configuración que permite al programa de variaciones que pueda operar solo con determinados conceptos, luego debe cumplir una condición lógica que solo puede ser considerada en la misma fórmula a través de la asignación “lVariac:=.t.”, Ejemplo:

```
IF CONDICION="A" .AND.;  
FORMA_PAG="E" .AND.;  
SALARIO>=1000  
lVariac:=.T. // Califica para aceptar variaciones  
nResult:=VARIAC*SALARIO  
ENDIF
```

VALIDACIÓN DE SINTAXIS

Es realizado por el sistema antes de efectuar la grabación del registro. El contenido del programa o fórmula es analizado sintácticamente, en caso de no cumplirlas, el sistema emitirá un mensaje indicando la frase que está incorrecta. No es posible detectar errores lógicos debido a que estos sólo se generan durante la ejecución de la fórmula. Si durante el proceso de cálculo “Nómina o Prenómina” es detectado algún error, será mostrado el concepto que lo genera y automáticamente queda excluido para seguir ejecutándose en los demás trabajadores. Posteriormente debe reversar la nómina, realizar las correcciones y preferiblemente ejecutar prenómina para verificar su correcto funcionamiento.

COMPONENTES PARA LA DEFINICION DEL CONCEPTO

- **Constantes:** Se emplean a través de la función CNS (nnn) y logran dinamizar sus resultados para ser asociados con la tabla de valores constantes que pueden cambiar en cualquier momento, logrando así afectar los resultados evitando modificar la fórmula.

Ejemplo:

Utilice una constante para asignar el % del Seguro social.

▪ **Funciones:** El sistema cuenta con un amplio conjunto de funciones dise1adas cuidadosamente para asistir y facilitar el dise1o de las f6rmulas, desde el c1culo de fechas, meses, d1as, consulta de hist6ricos, entre otras. En las pr6ximas secciones de este manual puede encontrar todas las funciones y una breve explicaci6n de uso. Tambi6n, en la parte inferior del editor de textos est1 disponible el bot6n "Funciones" que muestra el cat1logo de funciones con su respectiva sintaxis m1s un ejemplo de su utilizaci6n.

USO DE LOS VALORES DESDE OTRAS TABLAS

Cada tabla est1 representada a trav6s de un objeto que manipula un cursor SQL, se identifica con el nombre de la tabla y precedido de la letra "o" que significa "OBJETO", quedando as1 "oTabla", para emplear el nombre de algunas tablas relacionadas debe utilizar los siguientes nombres:

oNMDPTO	Departamentos
oNMCARGOS	Cargos
oNMUNDFUC	Unidad Funcional
oNMJORNADAS	Turnos o Jornadas
oNMBANCOS	Cuentas Bancarias
oNMGRUPO	Grupos
oNMPROFESION	Profesi6n
oNMESTADOS	Estados
oNMTABVAC	Tabla de Vacaciones
oNMAUSENCIA	Ausencias
oNMPAISES	Pa1ses
oNMTABLIB	Tabla de Liquidaci6n

Todas las nuevas tablas agregadas en el diccionario de datos y enlazadas con la tabla de trabajadores tambi6n estar1n disponibles para su uso bajo este mismo esquema.

Para utilizar el valor de un campo de una tabla relacionada debe ser indicado como sufijo de la tabla, como lo indica la siguiente sintaxis "oTabla:Campo", ejemplo: Para acceder al valor del turno de la mañana del día lunes de la jornada laboral debe emplear: "oNMJORNADAS:JOR_LUAM", para mayor comodidad utilice el botón "Campos" que se encuentra en la parte inferior del editor de textos, éste presenta un asistente de selección que facilita seleccionar campos de diversas tablas con sintaxis incluida.

Otras tablas relacionadas con el trabajador son enlazables a través de funciones específicas debido a la naturaleza y el objetivo de la misma, podemos citar: TABLAVAC(), TABLALIQ(), ACUMC_FCH() y demás funciones que acceden a los históricos de pagos.

Para acceder a cualquier tabla de datos o realizar una consulta personalizada, ADAPTAPRO provee el uso de la función OpenTable() que genera un cursor obtenido de una consulta SQL. Ver funciones para SQL.

USO DE VARIABLES

El sistema provee un conjunto de variables que pueden ser empleadas en el diseño de conceptos, pueden ser utilizadas en bifurcaciones o cálculos. Están compuestas por tres grupos: Nómina, Conceptos, Variaciones.

1. Variable del Objeto Nómina: Se refiere a los valores que encontrados en el motor de objeto "TNÓMINA" se encarga de generar todos los procesos de nómina. Este objeto está representado a través de la variable "oNm". Cabe destacar que cada ejecución de nómina genera una instancia, por esta razón este objeto mantiene los valores referidos a la nómina que se solicita. Estos son:

oNm:cOtraNom	Otra Nómina en Ejecución
oNm:cTipoNom	Tipo de Nómina
oNm:dDesde	Fecha de Inicio del Periodo de Pago
oNm:dHasta	Fecha Hasta del periodo de Pago
oNm:dDesdeAnt	Fecha Desde del Periodo Anterior
oNm:dhastaAnt	Fecha Hasta del Periodo Actual
oNm:lActualiza	Indica que se ejecuta Actualizar Nómina
oNm:lPrenomina	Indica que se ejecuta Prenómina
oNm:nDias	Días Trabajados Calculado en los Conceptos
oNm:nBaseA	Asignaciones y Deducciones para Salario A
oNm:nBaseB	Asignaciones y Deducciones para Salario B

oNm:nBaseC	Asignaciones y Deducciones para Salario C
oNm:nBaseD	Asignaciones y Deducciones para Salario D

2. Variables de Conceptos: Son Valores generados exclusivamente para la ejecución de cada concepto. También incluye el uso de variables utilizadas en los productos antecesores, con la finalidad de reconocer el uso de sus fórmulas. Estos valores son:

VARIAC	Valor Incluido en la tabla de Variaciones
NFACTOR1	Valor del factor1 que debe ser Almacenado en la Tabla de Observaciones
NFACTOR2	Valor del factor2 que debe ser Almacenado en la Tabla de Observaciones
NFACTOR3	Valor del factor3 que debe ser Almacenado en la Tabla de Observaciones
NFACTOR4	Valor del factor4 que debe ser Almacenado en la Tabla de Observaciones
ISR ASIGNA	Asignaciones configuradas para calcular ISLR
ASIGNA	Total, Asignaciones por Trabajador
VAROBSERV	Línea de Observación.
VARMEMO	Valor de Tipo Texto o Memo
DEDUCC	Total, Deducciones por Trabajador
FCHDESDE	Fecha Inicio del periodo de Cálculo (Emula Versión DOS)
FCHHASTA	Fecha Final del periodo de Cálculo (Emula Versión DOS)
TIPO NOM	Tipo de Nómina que está en Ejecución
OTRA NOM	Otra Nómina que está en Ejecución.
C_Xnnn	Valor de cada concepto ejecutado, la lista de estas variables depende de la cantidad de conceptos existentes, el Valor [X] Representa el primer carácter del concepto, y [nnn] completa el código que lo identifica, ejemplo: C_A001 representa el valor obtenido de la ejecución de la fórmula A001.

3. Variables del Sistema: Son valores generales para todo el sistema, se accede a través del objeto "oDp" y se encarga del manejo de valores de tipo público.

oDp:cEmpresa	Nombre de la Empresa.
oDp:dFecha	Fecha del Sistema.
oDp:dFchIni	Fecha Inicial del Ejercicio.

oDp:dFchFin	Fecha Final del Ejercicio.
oDp:cTipoNom	Nómina Seleccionada
oDp:cOtraNom	Otra Nómina Seleccionada
oDp:cCodTraIni	Código de Trabajador Inicial
oDp:cCodTraFin	Código de Trabajador Final
oDp:cCodGru	Código de Grupo
oDp:dDesde	Periodo Inicial de Pago
oDp:dHasta	Periodo Final de Pago
oDp:dDesdeAnt	Periodo Inicial Anterior al Periodo (oDp:dDesde)
oDp:dHastaAnt	Periodo Hasta Anterior al Periodo (oDp:dHasta)

Se puede notar que los valores para: Tipos de nómina y fechas se encuentran en los tres grupos de variables, aquí la diferencia:

Nómina	Se emplea cuando se ejecuta el proceso de nómina.
Conceptos	Mantienen compatibilidad con los sistemas de nómina predecesores.
Sistema	Se emplea para trasladar valores entre formularios, guardar y rescatar valores en el Dataset y se emplean en la definición y ejecución de reportes. Ver: Valores iniciales para rangos y Criterios.

Botones de herramientas:

En la parte inferior del editor de textos para los conceptos, el sistema provee diferentes opciones a través de botones para facilitar el diseño de la fórmula

Constantes: Presenta el catálogo de Constantes registradas, cada una muestra: Código, Descripción y Valor.



Este botón copia en clipboard el valor de la constante incluyendo sintaxis: CNS (nnn). Luego puede ser copiado dentro de la fórmula con la opción "Paste".



Realiza búsquedas en la tabla de las Constantes.



Cierra la tabla de las constantes.



Cód.	Descripción	Tipo	Valor	Alterado
001	% Retención del S.S.O.	Numérico	4.00000	<input type="checkbox"/>
002	% Seguro de Paro Forzoso	Numérico	0.50000	<input type="checkbox"/>
003	% Ley de Política Habitacional	Numérico	1.00000	<input type="checkbox"/>
004	Edad tope L.P.H (Masculino)	Numérico	60.00000	<input type="checkbox"/>
005	Edad tope L.P.H (Femenino)	Numérico	55.00000	<input type="checkbox"/>
007	Bono diario Alimento y Transporte	Numérico	0	<input type="checkbox"/>
008	Bono Subsidio por día	Numérico	0	<input type="checkbox"/>
010	Horas máximas de trabajo Semanal	Numérico	44.00000	<input type="checkbox"/>

Importar: Permite copiar la formula de un concepto a otro concepto.



Este botón copia el valor del concepto.



Realiza búsquedas en la tabla de Importar fórmulas del concepto.



Cierra la tabla de Importar.

Legislación: Permite visualizar e imprimir los artículos de la Ley Orgánica del Trabajo.



Graba el registro de un nuevo artículo de la Ley Orgánica del Trabajo.



Presenta el formulario de impresión del reporte del artículo de Ley del Trabajo.



Cierra la tabla de lo artículos de la Ley del Trabajo.

Funciones: Presenta el catálogo de funciones indicando la sintaxis y descripción. Adicional presenta un ejemplo.



Este botón copia en clipboard el valor de la función. Luego puede ser copiado dentro de la fórmula con la opción "Paste".



Realiza búsquedas en la tabla de Funciones de Nómina.



Graba los Cambios en la tabla de Funciones de Nómina.



Presenta el formulario de impresión del reporte de Funciones de Nómina.



Cierra la tabla de Funciones de Nómina.

Campos: Presenta el catálogo de Campos registrados, estos son: Trabajadores (Apellido, Banco), Lista de Bancos (BAN_BCOTXT, BAN_CODBCO), Cargos (CAR_CODIGO, CAR_SUELDO), Departamentos (DEP_DESCRI, DEP_ENCARG), etc. Cada uno muestra: Código, Tipo, Longitud, Decimales y Descripción.



Este botón copia en clipboard el valor del Campo.



Realiza búsquedas en los Campos de las Tablas.



Cierra la tabla de Campos.

Operadores: Presenta los diferentes operadores, tales como: Aritmético (+-*/), lógico (.AND. OR. NOT.) y Racional (=><).



Este botón copia en clipboard el valor de los operadores. Luego puede ser copiado dentro de la fórmula con la opción "Paste".



Realiza búsquedas en la tabla de los operadores.



Cierra la tabla de operadores.

Variables: Presenta el valor de las diferentes variables, estas son: Nómina (oNm:cOtraNom, oNm:cTipoNom), Concepto (VARIAC, NFACTOR1), Sistema (oDp:cEmpresa, oDp:dFecha), Tabla de Vacaciones (TABDESDE, TABHASTA), Tabla de Liquidación (LIQCAUSA, LIQCOTRA).



Este botón copia en clipboard el valor de las Variables. Luego puede ser copiado dentro de la fórmula con la opción "Paste".



Realiza búsquedas en la Tabla de Variables.



Cierra la tabla de Variables.

CONFIGURACIÓN

Configuración

Presenta cinco (5) secciones gráficamente visibles en recuadros, cada uno se refiere a los diversos usos del concepto.

Configuración para Variaciones: Se refiere a las exigencias que debe cumplir el concepto para participar en el formulario de variaciones, es decir; sólo participan los conceptos que posean información en los campos: Expresión y Mensaje.

Representación: Permite identificar la medida o expresión del valor introducido por el usuario para la variable de concepto "VARIAC", este puede ser: Horas, Días, Dinero, Minutos, o cualquier referencia que describa la variación. Ejemplo, si el concepto está referido a horas extras y debe ser introducida la cantidad de horas trabajadas es necesario indicar "Horas" para este campo, así en el recibo de pago se puede especificar el origen del monto de la asignación que el trabajador está percibiendo. Igualmente si se trata de alguna deducción por descuentos en Bolívares, puede indicarse el signo "Bs." ó la palabra "Monto".

Mensaje para variación: Utilizado en el formulario de variaciones con la finalidad de indicarle al usuario el tipo de dato que debe introducir como valor de variación del concepto "VARIAC".

Nómina Vertical: Permite definir la etiqueta o el nombre del concepto con el cual se visualizará en la nómina vertical, está compuesto por dos campos:

Columna: Solicita el nombre con el cual se visualizará el concepto en la nómina vertical, en el caso que este vacío el concepto se visualizará con el código del mismo.

Variación: Solicita el nombre con el cual se visualizará la variación del concepto en la nómina vertical.

Tipos de Nómina: Contiene cuatro (4) casillas para marcar los diferentes tipos de nómina “Semanal, Quincenal, Mensual u Otra”, donde será utilizado el concepto. De esta manera los procesos de cálculo, solo utilizarán los conceptos relacionados con el tipo de nómina que se utiliza.

Tipos de Salarios: Indica la participación del resultado generado por el concepto para calcular los salarios promedios definidos en la tabla de promedios mensuales, básicamente estos son: A, B, C y D. También son denominados: Básico, Integral, Utilidades y Vacaciones. Todos los promedios se generan durante el proceso de cálculo y son grabados directamente sobre la tabla de Salarios a través de la función GRABARSALARIO() ejecutada desde los mismos conceptos, ejemplo: “H353 Salario Básico del Mes”, y “H354 Salario Integral Diario (FINMES)”. Genera las variables: oNm:nBaseA, hasta oNm:nBaseD. Las funciones PROMEDIO (cTipo, dFecha), devuelve el tipo de salario promedio según el periodo solicitado.

Otros: Agrupa otros parámetros del concepto, estos son:

Genera Acumulados

, Solo está disponible para los conceptos cuyo primer dígito sea distinto al significado de: “Asignaciones”, “Deducciones”, “Funciones” e “Históricos”, permitiéndole al proceso de actualización de nómina incluir “Grabar” o excluir los resultados de estos conceptos en la tabla de históricos de pagos, No está disponible para los conceptos ordinarios “A, D, F y H” debido su naturaleza o uso. La tabla de históricos de pago se encuentra en la base de datos en ADAPTAPRO NOMINA en la tabla nmhistóricos.

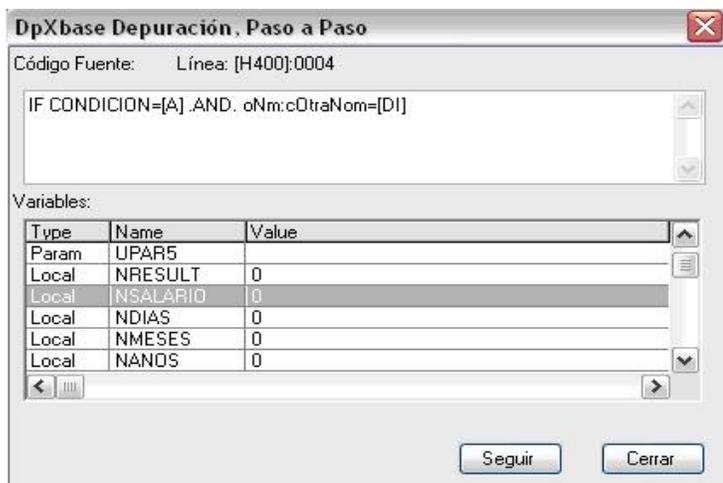
Acumula ISLR

Indica si el resultado del concepto es acumulado en la variable ISR_ASIGNA, necesario para calcular las deducciones del Impuesto sobre la renta.

Modo Depuración

Indica si el concepto activa el modo depuración durante el cálculo, bien sea para: Variación, Prenómina o Actualización de Nómina. El modo depuración presenta una ventana que visualiza pausadamente cada paso o línea que se ejecuta de la fórmula además permite visualizar todos los valores que interactúan con el concepto: Parámetros, Valores Locales, Privados y Campos de la ficha del trabajador. Los valores locales se refieren a las variables declaradas como “LOCAL”, los valores privados se refieren

a valores propios del proceso de nómina como: Tipo de Nómina, Asignaciones, Deducciones, Acumulados, Periodo, etc. El modo Depuración es altamente útil para revisar cuidadosamente los valores y ejecuciones de la fórmula logrando detectar las fallas o errores en corto tiempo.



COMENTARIOS

Comentarios

Presenta un editor de textos, que facilita registrar cualquier comentario o documentación sobre el diseño del concepto.

OTRAS CONSIDERACIONES

Los resultados generados por cada concepto obedecen a las fórmulas o instrucciones incluidas dentro de ella, forman parte de los datos cuya propiedad es del usuario, es decir; no forman parte del sistema, tampoco obedecen a criterios o interpretaciones jurídicas, por esta razón el licenciador no asume ningún tipo de responsabilidad por su contenido. Así mismo las fórmulas incluidas en calidad de ejemplo sólo son referencias de uso.

No emplee comillas simples (') en las fórmulas, debido a que estas pueden causar conflictos con la sintaxis SQL de la base de datos, emplee comillas dobles (") o corchetes ([]) para las expresiones de cadenas.

Los conceptos de Tipo Histórico no deben ser empleados para solicitar Variaciones, por tal motivo no figuran en el formulario "Variaciones".

BIFURCACION, OPERADORES LÓGICOS Y RELACIONALES:

En la mayoría de los conceptos han sido utilizados las bifurcaciones necesariamente para determinar comparaciones y estas se definen a través de la sentencia: IF/ELSE/ENDIF donde se ejecuta IF/ELSE si la comparación solicitada por IF se cumple de lo contrario será ejecutado ELSE/ENDIF, ejemplo:

```
IF <[Comparación]>  
... Instrucciones  
ELSE  
... Instrucciones  
ENDIF
```

La sentencia IF va acompañada de una expresión o multiexpresión que debe generar un resultado de tipo lógico "Verdadero .T. o Falso .F.) y éstas se construyen utilizando Lógicos (.AND. :OR. NOT) y/o relacionales (=,>,<,>=,<= y \$) Ejemplo, Trabajadores cuya condición es Activo o Liquidado, el campo es CONDICION y soporta un dígito.

```
IF (CONDICION="A" .OR. CONDICION="L")  
ELSE  
ENDIF
```

Operadores Lógicos: Se refieren a la evaluación lógica de una o varias expresiones que generan un resultado "Verdadero o Falso".

AND (Y), ejemplo, se requiere el cumplimiento de dos condiciones juntas.

```
IF CONDICION="A" .AND. SALARIO<=3500000  
...  
ELSE  
...  
ENDIF
```

OR (o Condicional), ejemplo, se requiere el cumplimiento de una de las dos condiciones.

```
IF CONDICION="A" .OR. SALARIO<=350000  
...  
ELSE  
...  
ENDIF
```

NOT (Negación), Niega el resultado de la condición y genera el valor contrario y se aplica sobre cualquier expresión lógica, ejemplo, si esta condición no se cumple:

```
IF NOT (CONDICION="A" .AND. SALARIO<=350000)
....
ELSE
....
ENDIF
```

MultiExpresión:

```
IF (nResult:=SALARIO*2, nSalario:=nSalario+100, nResult>0 .AND. CONDICION="A")
```

Operadores Relacionales: Como su palabra lo dice, es utilizado para relacionar valores del mismo tipo de datos como por ejemplo, determinar si un valor es igual o diferente que otro.

= Igual. Utilizado para realizar comparaciones primarias donde sólo se requiere comparar los primeros datos de las expresiones, ejemplo: "EXPRESION"="EXPRESIONES", devuelve verdadero.

== Exactamente Igual. Se diferencia de "=" en la comparación de valores, donde == evalúa la expresión o valor completamente mientras que "=" solo los primeros valores que coincidan, por tal motivo se recomienda utilizar "==". Ejemplo: "EXPRESION"=="EXPRESIONES", Devuelve .F.

¡= No Igual. Niega la expresión de igualdad, también puede escribirse .NOT. (nValor=nValor) o ¡nValor=nValor

<> Diferente.

>= Mayor o Igual.

<= Menor o Igual.

> Mayor que.

< Menor que.

\$ Cadena Asociada, Revisa si el contenido de una cadena está incluida en otra cadena, ejemplo:

IF "A"\$"ABCD", devuelve verdadero debido a que "A" se encuentra en la primera posición de "ABCD", por tal motivo se observa el uso de CONDICION\$"AL" en gran parte de los conceptos, que significa CONDICIÓN del Trabajador está dentro de la expresión "AL" activo o liquidado y considerando que la longitud de condición es un Carácter.

UTILIZACIÓN DE COMENTARIOS Y CAMPOS MEMOS O TEXTOS

Los resultados generados por los conceptos pueden ser acompañados de comentarios en dos formas: Una línea de cuarenta (40) caracteres representados a través de la variable "VAROBSERV"

y un campo de tipo memo o texto representado a través de la variable "VARMEMO". Ambas pueden ser construidas directamente en el mismo concepto con tan solo asignar el valor, ejemplo:

VAROBSERV:= "Antigüedad "+ANTIGÜEDAD(FECHA_ING)

Genera la expresión: Antigüedad 03a,0m,0d

VARMEMO:= "Inicio: "+DTC(TABDESDE)+CRLF+;

"Fin: "+DTC(TABHASTA)+CRLF+;

"Reintegro: "+DTC(TABFCHREI)

Genera un valor de texto compuesto por tres líneas y recordamos que CRLF es una declaración que indica retorno de Carro o salto de línea.

Durante el proceso de actualización, el valor de VAROBSERV es almacenado en la tabla NMOBSERV y se está vinculado con la tabla NMHISTORICO a través de los campos OBS_NUMERO y HIS_NUMOBS. También, la variable "VARMEMO", es almacenada en la tabla "NMMEMO" y se relaciona con la tabla NMHISTORICO a través de los campos "HIS_NUMMEM" y "MEM_NUMERO". Ambos valores pueden ser utilizados para el diseño de los recibos de pago.

Para emplear estos valores en la impresión de pre Nómina para crystal report, el sistema provee el uso de una tabla temporal que se describen a continuación:

TABLAS TEMPORALES PARA PRENÓMINA

PRENM.DBF Contiene los datos del resultado de la pre Nómina por cada concepto y trabajador

NMMONEDAS.DBF Contiene el nombre de cada moneda y total de monedas.

NMMEMO.DBF Almacena los valores de las observaciones y campos memos en una sola tabla. Estos valores son: VAROBSERV, VARMEMO, NFACTOR1, NFACTOR2, NFACTOR3 y NFACTOR4. La relación con la tabla PRENM.DBF se establece entre el campo HIS_NUMMEM con MEM_NUMERO de la tabla NMMEMO.DBF.

NMBILLETES.DBF Desglose monetario por trabajador, monto en letras y antigüedad laboral. Se relaciona con la tabla NMPRENM.DBF a través del campo "CÓDIGO" con el campo CODTRA de la tabla NMBILLETES.DBF

NMMONEDAS.DBF Contiene la denominación de cada billete o moneda y total de las mismas, necesarias para indicar el desglose por trabajador.

TABLAS TEMPORALES PARA LA IMPRESIÓN DE RECIBOS:

NMRECIBOS.DBF Contiene la consulta general de los recibos de pago, y ésta contiene información sobre la tabla de: Recibos, Trabajadores e Históricos.

NMBILLREC.DBF Desglose monetario por recibo, monto en letras y antigüedad laboral. Se relaciona con la tabla de recibos a través del campo "NUMREC", número de recibo de la tabla NMRECIBOS.DBF

NMOBSERV.DBF, Contiene todo la información referida a los valores: VAROBSERV, FACTOR1, FACTOR2, FACTOR3 y FACTOR4, relacionada con cada concepto de pago, éste se relaciona con la tabla NMRECIBOS.DBF a través del campo HIS_NUMOBS y OBS_NUMERO de la tabla NMOBSERV.DBF

NMMEMORE.DBF Contiene las observaciones registradas en campos memos o textos asociados con la tabla históricos de pago, se relaciona con la tabla NMRECIBOS.DBF a través del campo HIS_NUMREC y MEM_NUMERO de NMMEMORE.DBF

NMMONEDAS.DBF Contiene la denominación de cada billete o moneda y total de las mismas, necesarias para indicar el desglose por recibo.

Consideraciones:

El desglose monetario está disponible en las estructuras de datos para recibos y prenómina, solo se realiza la distribución para los recibos y/o trabajadores cuya forma de pago sea "EFECTIVO". Al final del reporte de nómina se desglosa el total de billetes o monedas necesarias para realizar el pago a todos los trabajadores.

CONFIGURACIÓN DEL CONCEPTO

Salario normal: Se refiere a aquellos conceptos derivados del salario normal sin ningún tipo de beneficios, cabe destacar que sólo pueden definirse como salario de la jornada ordinaria: Al tipo de nómina Semanal más días de descanso (Sábado y Domingo), Quincenal, Mensual más el pago de las vacaciones. También puede definirse como los pagos por jornadas sin otros beneficios. Ver función: **ASIGN_NORM()**.

CÁLCULO DE SALARIOS PROMEDIOS

El sistema posee una tabla denominada "Tabla de Salarios" compuesta por siete columnas: Código, Año, Mes y cuatro casillas para los salarios: "A,B,C y D", denominados por los conceptos sugeridos: Básico, Integral, Utilidades y Vacaciones.

Cada concepto de pago, posee cuatro casillas que permiten referenciar a los tipos de salario que afecta y de esta manera es posible establecer los criterios para determinar los diversos salarios por trabajador.

El proceso de pre-nómina y actualización de nómina contienen cuatro listas de conceptos donde cada una contiene los códigos de conceptos que afectan los salarios.

Básico: oNm:aConceptosA
Integral: oNm:aConceptosB,
Utilidades: oNm:aConceptosC
Vacaciones: oNm:aConceptosB

Conjuntamente con la función: ACUMC_FCH(<Conceptos>,<dDesde>,<dHasta>) es posible determinar el monto acumulado de un grupo de conceptos según la fecha del periodo de pago. Ésta

función es empleada para determinar el monto mensual según los conceptos para cualquier tipo de salario y para determinar el valor diario es necesario dividirlo entre los días del periodo de pago, para registrarlo se diseñó el concepto "H345 Días del periodo de pago" que almacena la cantidad de días del periodo de pago por cada trabajador, tomando en cuenta todas las fechas que inciden: Ingreso, Egreso, Inicio de Vacaciones, Fin de Vacaciones y Culminación de contratos.

De esta forma los conceptos poseen predefinidos las siguientes fórmulas:

H345: Almacena en cada periodo de pago los días Calendario trabajados.

H348: Al final del mes calcula el salario Diario para Vacaciones.

H349: Al final del mes calcula el salario Diario para Utilidades.

H353: Al final del mes calcula el Salario Diario básico.

H354: Al final del mes calcula el Salario Integral, que incluye la participación de los resultados generados por los conceptos: "H350 Apartado para Utilidades" que se aplica utilizando el salario calculado por el concepto H349. También el Salario Integral es afectado por el resultado del concepto "H352 Apartado Bono Vacacional".

Todos los conceptos emplean la función: GRABARSALARIO(<cTipoSalario>,<nValor>)

Graba en la tabla mensual de salario, por cada trabajador los resultados obtenidos en los conceptos de pago. Ejemplo:

GRABARSALARIO("B",nResult) Almacena el salario integral según la fecha Desde o Hasta del periodo de pago. La fecha empleada para determinar los periodos de pagos depende de la configuración del sistema, puede ser la fecha "Desde" o "Hasta".

H350: Apartado para Utilidades, se calcula al final del mes, utiliza el salario calculado por el Concepto H349, y su resultado es utilizado para determinar el salario Integral.

H352: Apartado para bono Vacacional, se calcula al final del mes, utiliza el salario básico calculado por la función "F003 Valor de Jornada Diaria", y su resultado es utilizado para determinar el salario Integral.

Gracias al salario integral calculado en el concepto H354 es posible construir el pasivo laboral en el concepto "H400 Compensación Mensual Antigüedad Art. 142".

Cabe destacar que la identificación del tipo de sueldo en cada concepto de pago permite definir y calcular los diferentes salarios, además: Las reservas para los futuros compromisos y pasivos laborales.

DESCRIPCIÓN DE ALGUNOS CONCEPTOS INCLUIDOS COMO EJEMPLOS

A200: Cálculo de Cesta ticket, éste obedece al tipo de "Otra Nómina" [CT], éste se determina según los días hábiles del mes, menos los días de inasistencia, multiplicado de la constante 90 "Valor diario del cesta ticket". Recuerde que los días hábiles obedecen a la jornada o turno asociado al trabajador.

N010: Aumento de Salarios. Éste obedece al tipo de Nómina "Aumento de Salario", depende del monto incluido en la variación "Monto que se desea Incrementar", el proceso de actualización reemplaza el nuevo sueldo según el anterior más el incremento, el sueldo anterior es almacenado para luego poder ser reestablecido en caso de ser ejecutado el proceso de reversión.

CONSTANTES

Permite ingresar valores fijos que se utilizaran en los cálculos de los conceptos para todos los trabajadores, dichos valores son editables, es decir, que pueden ser modificados cuando la empresa lo requiera y automáticamente el concepto toma el nuevo valor. Ejemplo; el valor o el porcentaje de recargos para pagos de horas extras, días de utilidades, vacaciones, etc.

Importante: El sistema ofrece en calidad de **EJEMPLO** la tabla llena con todas las constantes, estas pueden ser complementadas o adaptadas a los requerimientos del usuario.

JORNADAS Y TURNOS

Representa las definiciones de horario o Jornada laboral que debe ser asociada con cada trabajador, necesario para determinar la cantidad de horas exigidas por el concepto: "H006 Días Hábiles Trabajados" y calculadas por la función HORAS_TRAB() basada en la jornada vinculada con el trabajador.

A través del editor de registros presenta cada uno de los turnos y jornadas definidos para el cálculo de horarios. Cada Jornada o Turno representa el horario de trabajo semanal definido diariamente para antes del meridiano “AM” y después del meridiano “PM” con su respectiva hora de Entrada y Salida.

El formulario de registro del trabajador exige el código de jornada que el trabajador debe cumplir, por tal motivo es necesario poseer definido las diferentes alternativas de horario. El proceso de instalación del sistema crea en forma automática seis (6) jornadas identificadas para cada tipo de nómina: “S,Q,M,O, AM y PM”. “AM” y “PM” se refieren a jornadas para medio tiempo laboral de cuatro (4) horas diarias desde el lunes hasta el viernes. Para las jornadas “S,Q,M y O” se define de lunes a viernes actividades de ocho (8) horas diarias de 08:00am hasta 12:00am y 02:00pm hasta las 06:00pm.

La función de Nómina JOR_SEMHRS(), determina la cantidad de horas semanales definidas para un modelo de jornada o turno.

Importante: El sistema ofrece en calidad de **EJEMPLO** la tabla llena con los turnos, estas pueden ser complementadas o adaptadas a los requerimientos del usuario.

DÍAS FERIADOS

Este formulario permite ingresar los días feriados, cuyo valor se utilizará en los cálculos de los conceptos de pago. Si se activa el parámetro de los días de Carnavales y Semana Santa en configuración de empresa, pestaña Empresa, no hace falta registrarlo en este formulario.

Importante: El sistema ofrece en calidad de **EJEMPLO** la tabla llena con todos los días feriados, estas pueden ser complementadas o adaptadas a los requerimientos del usuario.

OTRAS NÓMINAS

Permite ingresar o registrar otras nóminas utilizadas por la empresa, diferentes a las nóminas normales (Semanal, Quincenal y Mensual). Por ejemplo Vacaciones, Utilidades, Guardería, etc.

Si el usuario crea “Otra Nómina”, tiene que tomar en cuenta, que ésta por si sola no arroja ningún valor, para ello también tiene que crear los conceptos necesarios.

Importante: El sistema ofrece en calidad de **EJEMPLO** la tabla llena con todas las otras nóminas, estas pueden ser complementadas o adaptadas a los requerimientos del usuario.

CUENTAS CONTABLES

Permite ingresar o registrar las cuentas contables “**Plan de cuentas propietario**”. Es importante destacar que si ADAPTAPRO NOMINA está integrado con el Administrativo y Contabilidad, no es necesario registrar el plan de cuenta, ya que automáticamente tomará el que está registrado en Contabilidad “**Plan de cuentas externo**”.

Importante: El sistema ofrece en calidad de **EJEMPLO** la tabla con cuentas contables para el caso que no trabaje integrado, estas pueden ser complementadas o adaptadas a los requerimientos del usuario.

TIPOS DE AUSENCIAS

Permite ingresar o registrar los tipos de ausencias relacionados con el trabajador, define la aplicación en descansos, feriados, laborables y reposo del I.V.S.S.

Importante: El sistema ofrece en calidad de **EJEMPLO** la tabla llena con los tipos de ausencias, estas pueden ser complementadas o adaptadas a los requerimientos del usuario.

TASAS DE INTERESES

Permite registrar las tasas de interés para el cálculo de los intereses sobre prestaciones, a través de los campos fechas, prestaciones, banca, préstamos y otros.

CARTA PARA EL TRABAJADOR

Permite ingresar todas las cartas necesarias, dichas cartas tienen que estar guardadas previamente en formato Word y en la dirección ADAPTAPRO NOMINA\cartas. Las cartas serán utilizadas en el formulario del trabajador por medio del botón “Emisión de cartas”.

UNIDAD TRIBUTARIA

Permite ingresar el monto de la Unidad Tributaria por años. Dicho monto será utilizado para calcular el ARI del trabajador. El formulario solicita la siguiente información: fecha de vigencia, monto de la Unidad Tributaria, Gaceta y Fecha de la Gaceta.

TARIFA DE I.S.L.R. (ART. 50)

En este formulario se registra la tarifa N. 1, se1alada en el art6culo 50 de la Ley de Impuesto Sobre La Renta, la misma es utilizada para calcular el ARI por trabajador. El formulario solicita la siguiente informaci3n: Hasta, en la cual se colar1 el tope de cada tramo, el porcentaje de retenci3n y el sustraendo.

FUNCIONES DE N6MINA

FECHAS Y CALENDARIO

ANIVERSARIO()	<p>Determina la fecha Data est1 de aniversario</p> <p>Sintaxis :ANIVERSARIO(<dFecha>,<dDesde>,<dHasta>)</p> <p>Devuelve .T. o .F. Si la fecha de ingreso es aniversario dentro del periodo de pago.</p> <p>Ejemplo:</p> <p>//</p> <p>ANIVERSARIO("15/01/2002","01/01/2004","15/01/2004")</p> <p>Resultado: .T. (15 de enero esta dentro del periodo de pago)</p>
ANTIGUEDAD()	<p>Determina la cantidad de D6as, Meses y A1os transcurridos entre dos fechas.</p> <p>Sintaxis:</p> <p>ANTIGUEDAD(<dDesde>,<dHasta>,@nAnos,@nMeses,@nDias0)</p> <p>Devuelve en una cadena: La antigüedad de la fecha, adem1s calcula los a1os en "nAnos", los meses en "nMeses" y los d6as en "nDias", los par1metros: nDias, nMeses y nAnos deben pasar como referencias por ello requiere el s6mbolo "@"</p> <p>Ejemplo:</p> <p>//</p> <p>LOCAL nAnos:=0,nMeses:=0,nDias:=0</p>

	<p>ANTIGUEDAD("15/01/1998","10/09/2004",@nAnos,@nMeses, @nDias</p> <p>Resultado: "06a,08m,05d" y las variables: nAnos=6, nMeses=8 y nDias=5</p>
CARNAVAL()	<p>Sintaxis :CARNAVAL(dFecha)</p> <p>Determina el lunes de Carnaval correspondiente al año indicado en el argumento <dFecha>. Si el Argumento <dFecha> es Omitido, es asumida la fecha del sistema <oDp:dFecha Ejemplo:</p> <p>//</p> <p>CARNAVAL()</p> <p>Resultado : 07/02/2005</p> <p>Para determinar el día martes de Carnaval, puede utilizar: CARNAVAL()+1</p>
DIAFERIADO()	<p>Sintaxis :DIAFERIADO(Fecha)</p> <p>Determina Si La fecha Especificada está registrada en el Calendario de días feriados. Devuelve un valor Lógico. .T. Para verdadero y .F. para falso.</p> <p>Ejemplo:</p> <p>//</p> <p>DIAFERIADO("01/05/95")</p> <p>.T. El resultado fue verdadero, La fecha especificada, se encuentra en el calendario de días feriados.</p>
DIASFERIAD()	<p>Sintaxis :DIASFERIAD(FchInicio,FchFin)</p> <p>Obtiene la cantidad de días feriados entre dos fechas y que estén registrados en el calendario de días feriados. Este cálculo no considera los días de la Jornada Laboral.</p> <p>Ejemplo:</p> <p>//</p> <p>DIASFERIAD("25/04/93","01/05/93")</p> <p>Resultado: 1 porque "01/05/93" es sábado y esta registrado en el calendario de días feriados.</p>
DIAS_HAB()	<p>Sintaxis: DIAS_HAB(FchInicio,FchFin)</p> <p>Obtiene los días hábiles transcurridos entre dos fechas, Los días hábiles</p>

	<p>pueden estar afectados por d1as Feriados.</p> <p>Ejemplo: // DIAS_HAB("26/04/92","02/05/92")</p> <p>Resultado: 4 d1as h1biles, porque (27/Lunes, 28/Martes, 29/Mi3rcoles, y 30/Jueves) son h1biles, (01/Viernes) "Seg1n calendario" es feriado y 26/Domingo y 02/S1bado No son H1biles. "Seg1n la jornada Laboral".</p>
<p>DIAS_NOHAB()</p>	<p>Sintaxis: DIAS_NOHAB(FchInicio,FchFin)</p> <p>Obtiene los d1as No h1biles transcurridos entre las dos fechas especificadas. Estos d1as no h1biles est1n sujetos a las fechas del calendario "Feriados o Fuera de la Jornada Laboral".</p> <p>Ejemplo: // DIAS_NOHAB("26/04/92","02/05/92")</p> <p>Resultado: 3 d1as No H1biles. Seg1n el calendario y los d1as Feriados.</p>
<p>DIAS_TRAB()</p>	<p>Sintaxis :DIAS_TRAB(Nd1as)</p> <p>Determina la cantidad de d1as activos que tiene el trabajador durante el per1odo de pago en proceso, Tomando en cuenta la Fecha de Ingreso, Inicio y Culminaci3n de las vacaciones y Fecha de Egreso.</p> <p>Ejemplo: // DIAS_TRAB(5) //Resultado 4 Si el trabajador inicio labores el d1a martes los d1as activos trabajados son: Mar, Mie, Jue, Vie (S1bado y domingo: depende de la jornada laboral).</p>
<p>FCHANUAL()</p>	<p>Sintaxis: FCHANUAL(FechAnterior,FechActual)</p> <p>Determina una fecha anterior con respecto a una fecha actual, Siendo 1til para determinar en que fecha del a1o actual corresponder1a la fecha del a1o anterior.</p>

	<p>Ejemplo: // FCHANUAL("05/01/92","30/10/95") Resultado: 05/01/92, Será la Fecha actual Iniciada por 05/01/92</p>
<p>FCHHABIL ()</p>	<p>Sintaxis: FCHHABIL (FechAnter, Ndias) Determina según la cantidad suministrada en días, cual es la próxima fecha hábil, Tomando en cuenta los días feriados del calendario a partir de la fecha especificada. Ejemplo: // FCHHABIL ("01/12/95",21) Resultado: 02/01/96, Retorna la fecha hábil transcurridos 21 días.</p>
<p>FERIAD_DOM()</p>	<p>Sintaxis: FERIAD_DOM(FchInicio,FchFin) Determina la cantidad de domingos transcurridos entre las dos fechas especificadas y que figuran en el calendario como días feriados. Ejemplo: // FERIAD_DOM("01/01/95","07/01/95") Resultado: 1</p>
<p>FERIAD_SAB()</p>	<p>Sintaxis: FERIAD_SAB(FchInicio,FchFin) Determina la cantidad de sábados transcurridos entre las dos fechas especificadas y que figuran en el calendario como días feriados. Ejemplo: // FERIAD_SAB("25/04/93","01/05/93") Resultado: 1</p>
<p>FINMES()</p>	<p>Sintaxis: FINMES (CriterioFecha)</p>

	<p>Determina si la fecha del período de pago corresponde a la última semana del mes, tomando las fechas "FchDesde" y "FchHasta". Esta Función devuelve un valor lógico, si es verdadera devuelve .T. de lo contrario devolverá .F.</p> <p>Ejemplo: // FINMES ()</p> <p>El resultado dependerá de la fecha del período de pago.</p>
FCHSIGMES(dFecha)	<p>Sintaxis: FCHSIGMES(dFecha)</p> <p>Determina la nueva fecha en el mes siguiente en el mismo número del día.</p> <p>Ejemplo: // FCHSIGMES("01/01/2004")</p> <p>Resultado: 01/02/2004</p>
FCHINIMES(dFecha)	<p>Sintaxis: FCHINIMES(dFecha)</p> <p>Determina el primer día del mes según la fecha solicitada.</p> <p>Ejemplo: // FCHINIMES("26/02/2004")</p> <p>Resultado: 01/02/2004</p>
FCHFINMES(dFecha)	<p>Sintaxis: FCHFINMES(dFecha)</p> <p>Determina el último día del mes según la fecha solicitada</p> <p>Ejemplo: // FCHFINMES("26/02/2004")</p> <p>Resultado: 29/02/2004</p>
MESES ()	<p>Sintaxis: MESES(FchInicio,FchFin)</p> <p>Calcula la cantidad de Meses transcurridos entre dos fechas especificadas, dando como resultado un valor entero. Esta función no toma en cuenta ningún dato del trabajador y procesa días exactos, es decir toma en cuenta los días de</p>

	<p>las fechas y genera resultados de meses exactos.</p> <p>Ejemplo: // MESES ("01/01/1994","01/03/1995") Resultado: 2</p>
NLUNES()	<p>Sintaxis :NLUNES(FchInicio,FchFin)</p> <p>Determina la cantidad de lunes Transcurridos entre las dos fechas especificadas. Este Proceso no necesita la intervenci3n del calendario de d3as feriados, ni la Jornada Laboral.</p> <p>Ejemplo: // NLUNES("01/10/95","31/10/95") Resultado: 5 (lunes en el mes de octubre del 95)</p>
QUINCE()	<p>Sintaxis :QUINCE()</p> <p>Determina si el per3odo de pago corresponde a la primera quincena del mes, esto quiere decir que la variable "FchHasta" debe ser menor 3 igual que el d3a 15.</p> <p>Ejemplo: // QUINCE() Resultado: .F. si el per3odo de pago es "01/10/95 al 15/10/95".</p>
NDIAS()	<p>Sintaxis :NDIAS(FchInicio,FchFin,DjaSemana)</p> <p>Calcula cuantos d3as especificos han transcurrido entre dos fechas, indicando previamente cual de los d3as de la semana se va a contar, tomando en cuenta al domingo como primer d3a de la semana y al s3bado como 3ltimo.</p> <p>Ejemplo: //</p>

	<p>NDIAS("01/05/2000","31/05/2000",1)</p> <p>Resultado: 4 (domingos en el mes de mayo del 2000)</p>
NDOM()	<p>Sintaxis :NDOM(FchInicio,FchFin)</p> <p>Determina la cantidad de domingos Transcurridos entre las dos fechas especificadas. Este Proceso no necesita la intervenci3n del calendario de d3as feriados, ni la Jornada Laboral.</p> <p>Ejemplo: // NDOM("01/10/95","31/10/95")</p> <p>Resultado: 5 (domingos en el mes de octubre del 95)</p>
NMAR()	<p>Sintaxis :NMAR(FchInicio,FchFin)</p> <p>Determina la cantidad de martes Transcurridos entre las dos fechas especificadas. Este Proceso no necesita la intervenci3n del calendario de d3as feriados, ni la Jornada Laboral.</p> <p>Ejemplo: // NMAR("01/10/95","31/10/95")</p> <p>Resultado: 5 (martes en el mes de octubre del 95)</p>
NMIE()	<p>Sintaxis :NMIE(FchInicio,FchFin)</p> <p>Determina la cantidad de mi3rcoles Transcurridos entre las dos fechas especificadas. Este Proceso no necesita la intervenci3n del calendario de d3as feriados, ni la Jornada Laboral.</p> <p>Ejemplo: // NMIE("01/10/95","31/10/95")</p> <p>Resultado: 4 (mi3rcoles en el mes de octubre del 95)</p>
NJUE()	<p>Sintaxis :NJUE(FchInicio,FchFin)</p> <p>Determina la cantidad de jueves Transcurridos entre las dos fechas</p>

	<p>especificadas. Este Proceso no necesita la intervenci3n del calendario de d3as feriados, ni la Jornada Laboral.</p> <p>Ejemplo: // NJUE("01/10/95"."31/10/95") Resultado: 4 (jueves en el mes de octubre del 95)</p>
<p>NVIE()</p>	<p>Sintaxis :NVIE(FchInicio,FchFin)</p> <p>Determina la cantidad de viernes Transcurridos entre las dos fechas especificadas. Este Proceso no necesita la intervenci3n del calendario de d3as feriados, ni la Jornada Laboral.</p> <p>Ejemplo: // NVIE("01/10/95","31/10/95") Resultado: 4 (viernes en el mes de octubre del 95)</p>
<p>NSAB()</p>	<p>Determina la cantidad de s3bados Transcurridos entre las dos fechas especificadas. Este Proceso no necesita la intervenci3n del calendario de d3as feriados, ni la Jornada Laboral.</p> <p>Sintaxis: NSAB(FchInicio,FchFin)</p> <p>Ejemplo: // NSAB("01/10/1995","31/10/1995") Resultado: 4 (viernes en el mes de octubre del 95)</p>
<p>SEMANASANTA()</p>	<p>Sintaxis: SEMANASANTA(dFecha)</p> <p>Determina el jueves de Semana Santa correspondiente al a3o indicado en el argumento <dFecha>. Si el Argumento <dFecha> es Omitido es asumida la fecha del sistema <oDp:dFecha> Ejemplo: //</p>

SEMANASANTA()

Resultado : 24/03/2005

Para determina el día viernes Santo, puede utilizar: SEMANASANTA()+1

HISTÓRICOS

TOTAL()

Sintaxis: TOTAL(FchInicio,FchFin)

Permite obtener una sumatoria de todos los conceptos de asignación y deducción en el periodo solicitado.

Ejemplo:

//

TOTAL("01/01/95","31/12/95")

Resultado: Suma los conceptos de Asignación y Deducción en el año 95.

//

Sería idéntico solicitar en el mismo periodo las asignaciones mediante la función ASIGN(), las deducciones con DEDUC() y sumarlas.

ACUMC_MES()

Sintaxis: ACUMC_MES(Concepto,Fecha)

Devuelve el valor acumulado dentro del periodo mensual de cada uno de los conceptos actualizados. Según el tipo de concepto "A" o "D" dependerá el signo del resultado.

Ejemplo:

//

ACUMC_MES("A001","01/01/95")

Resultado: Devuelve el monto acumulado del concepto "A001" correspondiente a las nóminas procesadas en el mes de enero.

ACUMC_MES("A001")

Resultado: Dev. el monto total acumulado del concepto "A001" en enero.

ACUMF1_FCH ()

Sintaxis: ACUMF1_FCH (Concepto,FchInicio,FchFin)

Devuelve el valor acumulado del factor1 en cualquier período actualizado de

	<p>cada uno de los conceptos solicitados.</p> <p>Ejemplo:</p> <pre>// ACUMF1_FCH ("D014", "01/01/95", "31/12/95")</pre> <p>Resultado: Devuelve el monto acumulado del factor1 correspondiente al concepto "D014" de las n6minas procesadas del 01/01/95 al 31/12/95</p>
<p>ACUMF2_FCH ()</p>	<p>Sintaxis: ACUMF2_FCH (Concepto,FchInicio,FchFin)</p> <p>Devuelve el valor acumulado del factor2 en cualquier per6odo actualizado de cada uno de los conceptos solicitados.</p> <p>Ejemplo:</p> <pre>// ACUMF2_FCH("D014", "01/01/95", "31/12/95")</pre> <p>Resultado: Devuelve el monto acumulado del factor2 correspondiente al concepto "D014" de las n6minas procesadas del 01/01/95 al 31/12/95</p>
<p>ACUMF3_FCH ()</p>	<p>Sintaxis: ACUMF3_FCH (Concepto,FchInicio,FchFin)</p> <p>Devuelve el valor acumulado del factor3 en cualquier per6odo actualizado de cada uno de los conceptos solicitados.</p> <p>Ejemplo:</p> <pre>// ACUMF3_FCH ("D014", "01/01/95", "31/12/95")</pre> <p>Resultado: Devuelve el monto acumulado del factor3 correspondiente al concepto "D014" de las n6minas procesadas del 01/01/95 al 31/12/95</p>
<p>ACUMF4_FCH ()</p>	<p>Sintaxis: ACUMF4_FCH (Concepto,FchInicio,FchFin)</p> <p>Devuelve el valor acumulado del factor4 en cualquier per6odo actualizado de cada uno de los conceptos solicitados.</p> <p>Ejemplo:</p> <pre>// ACUMF4_FCH ("D014", "01/01/95", "31/12/95")</pre> <p>Resultado: Devuelve el monto acumulado del factor4 correspondiente al</p>

	concepto "D014" de las nóminas procesadas del 01/01/95 al 31/12/95
ACUMV_FCH()	<p>Sintaxis: ACUMV_FCH(Concepto,FchInicio,FchFin)</p> <p>Obtener el valor acumulado en las variaciones de los conceptos actualizados, dentro de un periodo de fechas dadas.</p> <p>Ejemplo: // ACUMV_FCH ("D001","01/01/95", "31/12/95")</p> <p>Resultado: Devuelve el monto acumulado del concepto "D001" (días de Inasistencias) correspondientes a las nóminas procesadas del "01/01/95" al "31/12/95".</p>
ACUMV_MES()	<p>Sintaxis: ACUMV_MES(Concepto,Fecha)</p> <p>Obtener el valor acumulado en las variaciones de los conceptos actualizados, en el mes.</p> <p>Ejemplo: // ACUMV_MES ("D001","01/01/95")</p> <p>Resultado: Devuelve el monto acumulado del concepto "D001" (días de Inasistencias) correspondientes a las nóminas procesadas el mes de enero.</p>
ASIGN ()	<p>Sintaxis: ASIGN (FchInicio,FchFin)</p> <p>Obtiene el total de las asignaciones del trabajador en curso, dentro de un periodo determinado.</p> <p>Ejemplo: // ASIGN ("01/01/95","28/02/95")</p> <p>Resultado: Realiza la sumatoria de todas las asignaciones en el periodo indicado para el trabajador en curso.</p>
ACUMULA_CLA(dDesde,dHasta,cCodCla)	<p>Obtiene desde la tabla histórica de Pagos, el monto acumulado según los conceptos asociados a la clasificación "cCodCla".</p> <p>Ejemplo:</p>

	<p>//</p> <p>ACUMULA_CLA("01/01/2004","31/12/2004","Utilidades")</p> <p>Obtiene el total de todos los pagos "Asignaciones y deducciones" relacionadas con conceptos asociados con la clasificación "Utilidades".</p>
<p>AVERAGE_CLA(dDesde,dHasta,cCodCla)</p>	<p>Determina el valor promedio relativo o Average a partir del monto obtenido de la consulta de la tabla histórico de pagos.</p>
<p>SALARIO(cTipo,dFecha)</p>	<p>Devuelve el salario histórico almacenado en la tabla "Sueldos Promedios", obtenido del proceso de actualización de nómina. El parámetro <cTipo> indica el tipo de sueldo "A","B","C" o "D", si el valor es omitido es asumido el valor "A", el parámetro <dFecha>, indica la fecha del año y mes solicitado</p> <p>Ejemplo:</p> <p>nSalario:=SALARIO("A",oNm:dDesde)</p>
<p>ACUMC_FCH ()</p>	<p>Sintaxis: ACUMC_FCH(Concepto,FchInicio,FchFin)</p> <p>Obtener por cada trabajador el resultado histórico almacenado por cada concepto actualizado del período solicitado. Según el tipo de concepto "A" ó "D" dependerá el signo del resultado.</p> <p>Ejemplo:</p> <p>//</p> <p>ACUMC_FCH ("A001","01/01/95","31/12/95")</p> <p>Resultado: Devuelve el monto acumulado del concepto "A001" correspondiente a las nóminas procesadas del 01/01/95 al 31/12/95.</p> <p>ACUMC_FCH("A001")</p> <p>Resultado: Devuelve el monto total acumulado del concepto "A001 Salario Semanal"</p>
<p>DEDUCC()</p>	<p>Sintaxis :DEDUCC(FchInicio,FchFin)</p> <p>Obtiene el total de las deducciones del trabajador en curso, dentro de un periodo determinado.</p> <p>Ejemplo:</p> <p>//</p> <p>DEDUCC("01/01/95","31/12/95")</p>

Resultado: Realiza la sumatoria de todas las deducciones almacenadas durante el a1o 95

TABLAS ASOCIADAS

<p>VARPRE()</p>	<p>Sintaxis: VARPRE(Concepto)</p> <p>Obtiene el Valor Num6rico almacenado en la variaci3n almacenada en otro concepto de N6mina.</p> <p>Ejemplo:</p> <p>//</p> <p>VARPRE("D001")</p> <p>Obtiene el valor de la variaci3n almacenada en el Concepto "D001 D1as de Inasistencia", que es requerida para efectuar otros c1culos que dependen de el valor de la variaci3n de tal concepto, como en los casos de los bonos, Depende tambi3n de las Inasistencias.</p>
<p>TABLA()</p>	<p>Sintaxis: TABLA (Tabla, Clave, Campo)</p> <p>Esta funci3n permite localizar y acceder a los datos almacenados en cualquier tabla predefinida en el sistema.</p> <p>Ejemplo:</p> <p>//</p> <p>TABLA ("0001",MESES(FECHA_ING,FCHHASTA),"PREAVISO")</p> <p>Resultado: Valor num6rico encontrado en el campo "PREAVISO" de la tabla 0001 dependiendo de las fechas suministradas.</p>
<p>CNS()</p>	<p>Sintaxis: CNS(Numero)</p> <p>Devuelve el valor almacenado en una constante de N6mina Previamente Definida. Ejemplo:</p> <p>//</p> <p>PORCEN (1000, CNS (01))</p> <p>Resultado: 40.</p> <p>La constante CNS(01) contiene el valor 4, que pertenece al % de retenci3n del</p>

	I.V.S.S y al Aplicarse el 4% al valor 1000 dará como resultado 40.
CREATABPRE()	<p>Sintaxis: CREATABPRES(nMonto,nCuota,nTasa,@cNumero)</p> <p>Crea automáticamente un registro en la tabla de préstamos "NMTABPRES" para el trabajador que está en proceso de actualización.</p> <p>El registro está asociado al código del trabajador, al número de recibo donde se otorga el préstamo, monto solicitado, valor de la cuota de pago y el % de la tasa de interés.</p> <p>nMonto:= Indica el monto del préstamo.</p> <p>nCuota:= Cuota fija de Pago.</p> <p>nTasa := % de Intereses que será aplicado al préstamo.</p> <p>@cNumero Recibe por referencia el número del registro de préstamo.</p> <p>Sintaxis:</p> <p>LOCAL nCuota :=VARPRE("N000")</p> <p>LOCAL nTasa :=VARPRE("N001")</p> <p>nResult:=CREATABPRES(VARIAC,nCuota,nTasa,@cNumero)</p> <p>RETURN nResult</p>
CREATABVAC ()	<p>Sintaxis: CREATABVAC (FchInicio,FchFin,Dias)</p> <p>Crea los registros para la tabla de vacaciones del trabajador en curso, según los parámetros de Fecha Inicio y Fin, disfrutando los días estipulados en Ndías.</p> <p>Ejemplo:</p> <p>Teniendo creadas las constantes x1 y x2 como fecha inicio y fin se logra:</p> <p>//</p> <p>CREATABVAC (CNS(x1),CNS(x2),DIAS_HAB(CNS(x1),CNS(x2)))</p> <p>El sistema tomará las constantes x1 y x2 como fecha de inicio y fin respectivamente y calculará los días hábiles calendario mediante la función DIAS_HAB()</p>
GRATABVAC()	Sintaxis: GRATABVAC(Campo,Contenido)

	<p>Permite modificar directamente los campos de la tabla de vacaciones sólo para los trabajadores que estén registrados en la misma. Estos campos son: TAB_DESDE, TAB_HASTA y TAB_DIAS.</p> <p>Ejemplo :</p> <pre>// GRATABVAC("TAB_DIAS",DIAS_HAB(TABDESDE,TABHASTA))</pre> <p>Modifica el campo TAB_DIAS el cual contiene los días a disfrutar, según los días hábiles contenidos entre las fechas TABDESDE y TABHASTA. Estas dos variables son globales y provienen de los campos TAB_DESDE y TAB_HASTA respectivamente, localizados en la tabla de vacaciones.</p>
<p>LEEPRESTAMO()</p>	<p>Sintaxis: LEEPRESTAMO(nCuota,@cMemo,lDeuda,lInteres)</p> <p>Lee desde la tabla de préstamos el monto que deben ser amortizado por el trabajador que está en proceso de pre Nómina y actualización de nómina.</p> <p><nCuota> Indica el monto máximo que puede ser descontado del total de los préstamos pendientes.</p> <p><@cMemo> Por referencia acepta la descripción que genera cada préstamo afectado.</p> <p><lDeuda> Indica con .T. si obtiene el total de deuda de todos los préstamos, necesario para el proceso de liquidación.</p> <p><lInteres> Indica con .T. si calcula el valor de los intereses por cada préstamo y lo adiciona en forma de texto sobre la variable <@cMemo>. En ningún momento el <lInteres> es sumado sobre el resultado de los préstamos.</p> <p>Ejemplo:</p> <pre>nResult:=LEEPRESTAMO(VARIAC,@VARMEMO,.F.,.F.)</pre> <p>Resultado de tipo numérico y depende del valor de los préstamos pendientes y <@VARMEMO> contiene el resumen de cada préstamo en forma enumerada.</p>
<p>TABLALIQ ()</p>	<p>Sintaxis: TABLALIQ ()</p> <p>Verifica la existencia del trabajador que está siendo procesado en la tabla de liquidación. Ésta función retorna un valor lógico booleano, .T. (verdad) si es encontrado o .F. (falso) si no es encontrado.</p> <p>Ejemplo:</p>

	<p>//</p> <p>TABLALIQ ()</p> <p>Resultado: Si el trabajador está en la tabla de liquidación retornará .T. o .F. en caso contrario.</p>
<p>TABLAVAC ()</p>	<p>Sintaxis: TABLAVAC ()</p> <p>Verifica la existencia del trabajador que está siendo procesado en la tabla de vacaciones. Esta función retorna un valor lógico booleano, .T. (verdad) si se encuentra o .F. (falso) si no se encuentra.</p> <p>Ejemplo:</p> <p>//</p> <p>TABLAVAC()</p> <p>Resultado: Si el trabajador esta en la tabla de vacaciones retornará .T. o .F. en caso contrario.</p>
<p>PERMISO()</p>	<p>Sintaxis :PERMISO(FchInicio,FchFin)</p> <p>Determina por cada trabajador la cantidad de días hábiles que han transcurrido durante el período de Permiso, Localizados entre las fechas especificadas.</p> <p>Ejemplo:</p> <p>//</p> <p>PERMISO("01/10/95","31/10/95")</p> <p>Resultado: Valor Numérico que contiene la cantidad de días hábiles para dicho proceso.</p>
<p>REPOSO ()</p>	<p>Sintaxis: REPOSO (FchInicio,FchFin)</p> <p>Determina por cada trabajador la cantidad de días hábiles que han transcurrido durante el período de reposo, localizados entre las fechas especificadas.</p> <p>Ejemplo:</p> <p>//</p> <p>REPOSO("01/10/95","31/10/95")</p> <p>Resultado: Valor Numérico que contiene la cantidad de días hábiles para dicho proceso.</p>

CARGAS_FAM()	<p>Sintaxis : CARGAS_FAM(cWhere,cCodTra)</p> <p>Determina las cargas familiares seg3n los datos almacenados en la tabla de familiares bajo la condici3n de "Dependencia", adem3s puede aceptar otras condiciones de b3squeda seg3n la cl3usula <cWhere>.</p> <p>Ejemplo:</p> <pre>// CARGAS_FAM("LEFT(FAM_PARENT,3)='Hij'")</pre> <p>Resultado: Devuelve la Cantidad de Hijos e Hijas bajo dependencia.</p>
NUM_DESCEN()	<p>Sintaxis : NUM_DESCEN(cWhere,cCodTra)</p> <p>Determina la cantidad de "Hijos" o "Hijas" registrados en la ficha familiar del trabajador. La cl3usula <cWhere> permite indicar otros filtros.</p> <p>Ejemplo:</p> <pre>// NUM_DESCEN("LEFT(FAM_PARENT,3)='Nie'")</pre> <p>Resultado: Devuelve la Cantidad de Hijos e Hijas y (Nietos o Nietas) relacionadas con el trabajador.</p>

FICHA DEL TRABAJADOR

Agrupar un conjunto de funciones para el dise1o de conceptos de pago y que emplean los datos directos del trabajador.

CONTAR ()	<p>Sintaxis: CONTAR (Condici3n)</p> <p>Obtiene la cantidad de trabajadores que cumplan con la condici3n dada. Si no se indica ninguna condici3n, ser3 calculado el n3mero total de trabajadores existentes en la tabla de trabajadores.</p> <p>Ejemplo:</p> <pre>// CONTAR("SALARIO = 5000")</pre>
------------------	---

	<p>Resultado: Número de casos cumplidos.</p> <p>CONTAR ("TIPO_NOM= 'S' ")</p> <p>Resultado: Cantidad de trabajadores registrados para nómina semanal.</p>
GRABARCAM()	<p>Sintaxis: GRABARCAM(<cCampo>,xContenido)</p> <p>Modifica el valor del campo <cCampo> por el valor de <xContenido> de la ficha del trabajador que está en proceso de actualización de nómina. Por razones de seguridad el valor previo del campo es almacenado en la tabla "NMGRABAR" identificado por el tipo de nómina y periodo que se actualiza con la finalidad de restaurar el valor del campo en caso de ser ejecutado el proceso de reversión.</p> <p>Ejemplo:</p> <pre>// GRABARCAM("CONDICION","L")</pre> <p>Modifica el campo <CONDICION> del trabajador en proceso de actualización, colocándole "L", esto le indicará al sistema que el trabajador ha sido liquidado. Igualmente es útil para el manejo de préstamos o acumulador de cualquier otro valor.</p>

SALARIOS

Ofrece un grupo de funciones diseñadas exclusivamente para generar valores para los diferentes tipos de salarios y obtener los diferentes salarios desde la tabla "NMRESTRA".

PROMEDIO_A(dFecha)	<p>Calcula y devuelve el Salario promedio de tipo A "Básico" para el mes solicitado en <dFecha>, el proceso es realizado a través del programa DpXbase "NMCALACUMT", también almacena el resultado en la tabla "NMRESTRA", en la casilla "A" para el mes y año indicado en <dFecha></p> <p>Ejemplo:</p> <pre>nSalario:= PROMEDIO_A(oNm:dHasta)</pre>
PROMEDIO_B(dFecha)	<p>Calcula y devuelve el Salario promedio de tipo "B" "Integral" para el mes solicitado en <dFecha>, el proceso es realizado a través del programa DpXbase "NMCALACUMT", también almacena el resultado en la tabla "NMRESTRA", en la casilla "B" para el mes y año indicado en <dFecha>. Las fórmulas "Ejemplos" emplean PROMEDIO_B en el concepto "H400 Antigüedad Art 142"</p>

	<p>Ejemplo:</p> <p>nSalario:= PROMEDIO_B(oNm:dHasta)</p>
PROMEDIO_C(dFecha)	<p>Calcula y devuelve el Salario promedio de tipo "C" "Utilidades" para el mes solicitado en <dFecha>, el proceso es realizado a través del programa DpXbase "NMCALACUMT", también almacena el resultado en la tabla "NMRESTRA", en la casilla "C" para el mes y año indicado en <dFecha>. Las fórmulas "Ejemplos" emplean PROMEDIO_C en el concepto "H350 Apartado para Utilidades"</p> <p>Ejemplo:</p> <p>nSalario:= PROMEDIO_C(oNm:dHasta)</p>
PROMEDIO_D(dFecha)	<p>Calcula y devuelve el Salario promedio de tipo "D" "Vacaciones" para el mes solicitado en <dFecha>, el proceso es realizado a través del programa DpXbase "NMCALACUMT", también almacena el resultado en la tabla "NMRESTRA", en la casilla "D" para el mes y año indicado en <dFecha>. Las fórmulas "Ejemplos" emplean PROMEDIO_D en el concepto "H351 Apartado para Vacaciones"</p> <p>Ejemplo:</p> <p>nSalario:= PROMEDIO_D(oNm:dHasta)</p>
PROMEDIO(cTipo,dFecha)	<p>Devuelve el Salario almacenado en la tabla "Salarios Promedios", según el periodo <dFecha> y la casilla "A","B","C" o "D" indicada en <cTipo>. No genera ningún cálculo, solo lee el valor de la tabla. Si <cTipo> no es enviado será asumido "A" y <dFecha> en caso de ser vacío asumirá la fecha "dHasta" del tipo de Nómina.</p>
SALARIOPRO ()	<p>Sintaxis: SALARIOPRO (MESES, FECHA)</p> <p>Calcula el salario promedio ponderado extraído desde la base de datos de salarios promedios mensuales que se generan por cada trabajador en cada proceso de actualización de la nómina.</p> <p>Ejemplo:</p> <p>//</p> <p>SALARIOPRO (3,"31/12/95")</p> <p>//Resultado: 1500,00 Suponiendo que el promedio de octubre es 1500,00 el de Noviembre 1000,00 y el de diciembre 2000,00. Entonces la ecuación sería (oct+nov+dic)/3</p>

<p>SALARIOULT ()</p>	<p>Sintaxis: SALARIOULT (Fecha)</p> <p>Extrae el último salario básico desde la base de datos de salario históricos por mes, que se generan por cada trabajador mediante el proceso de actualización de la nómina.</p> <p>Ejemplo: // SALARIOULT ("31/12/95")</p> <p>Resultado: Valor numérico extraído de la base de datos de salarios Históricos.</p>
<p>GRABAR SUELDO()</p>	<p>Sintaxis: GRABARSUELDO (nMonto)</p> <p>Guarda el Salario Mensual en el campo "RMT_SUELDO" en la tabla "NMRESTRA", que permite conocer por cada mes, el sueldo mensual por trabajador.</p> <p>Ejemplo: // GRABARSUELDO(SALARIO)</p> <p>Resultado: Devuelve el valor enviado como parámetro.</p>

INTERESES

<p>INTERESES()</p>	<p>Sintaxis: INTERESES(TabIntr,TabAntig,LosRet,Meses)</p> <p>Determina los intereses sobre el capital generado por los acumulados de antigüedad laboral por cada año cumplido en la empresa.</p> <p>Ejemplo: // INTERESES("002","003","A073","A087",CNS(63))</p> <p>//Resultado: Determinará los intereses a capital de prestaciones desde que el trabajador cumplió el primer año de servicio.</p>
---------------------------	--

INTERES()	<p>Sintaxis: INTERES(Tabl,Cap,AboRet,Fch1,Fch2,Neg)</p> <p>Determina los intereses sobre el capital generado por los acumulados de cualquier grupo de conceptos que conformen un capital por cada mes cumplido en la empresa.</p> <p>Ejemplo:</p> <p>//</p> <p>INTERES("002","H400","A410",NIL,FCHHASTA,NIL,.F.)</p> <p>Intereses sobre el concepto H400 (antigüedad) desde que el trabajador gener6 la primera transacci6n.</p>
FCHINITRI()	<p>Sintaxis: FCHINITRI(dFecha)</p> <p>Determina la fecha inicial del trimestre correspondiente al parámetro <dFecha>.</p> <p>Ejemplo:</p> <p>//</p> <p>FCHINITRI("28/02/2004")</p> <p>//Resultado: 01/01/2004</p> <p>Corresponde al primer trimestre del a1o, esta funci6n es 6til para determinar los rangos de reportes que se emiten por trimestres.</p>
FCHFINTRI()	<p>Sintaxis: FCHFINTRI(dFecha)</p> <p>Determina la fecha final del trimestre indicado en el parámetro <dFecha></p> <p>Ejemplo:</p> <p>//</p> <p>FCHFINTRI("28/02/2004")</p> <p>// Resultado: 31/03/2004 Corresponde a la fecha final del trimestre de la fecha 28/02/2004</p>
CREAVARCON()	<p>Sintaxis: FCHINITRI(dFecha)</p> <p>Determina la fecha inicial del trimestre correspondiente al parámetro <dFecha>.</p>

```
Ejemplo:
//
FCHINITRI("28/02/2004")
//Resultado: 01/01/2004
```

FUNCIONES XBASE

Explica brevemente el uso de funciones del l6xico estandar Xbase (Dbase, Clipper y FoxPro) para el manejo de datos: Num6ricos, L6gicos, Car6cter y Fechas. Todas las funciones indicadas en este segmento ya est6n incluidas dentro del programa ejecutable y han sido utilizadas para dise1nar las f6rmulas de los ejemplos:

<p>AT todo</p>	<p>Sintaxis: AT(cChar,cCadena)</p> <p>Determina la posici6n del o los caracteres especificados en <cChar>, de izquierda a derecha dentro de la secuencia de caracteres cadena<cCadena></p> <p>Ejemplo:</p> <p>AT("N","NOMINA")</p> <p>Resultado 1</p>
<p>EMPTY()</p>	<p>Sintaxis: EMPTY(uValor)</p> <p>Devuelve Verdadero .T. si el Valor de <uValue> est6 vaci6 y es aplicable a todos los tipos de datos xBase.</p> <p>Ejemplo:</p> <p>EMPTY("")</p> <p>Devuelve .T.</p> <p>EMPTY(0)</p> <p>Devuelve .T.</p> <p>EMPTY("N6mina")</p>

	Devuelve .F.
MIN()	<p>Sintaxis: MIN(dDesde1,dDesde2)</p> <p>Devuelve el menor de dos valores. Sólo puede ser utilizado para tipos de datos numéricos y fechas.</p> <p>Ejemplo:</p> <p>MIN(DTOC("01/01/2004"),DTOC("05/11/2010"))</p> <p>Resultado: 01/01/2004</p> <p>MIN(100,200)</p> <p>Resultado: 100</p>
MAX()	<p>Sintaxis: MAX(dFecha1,dFecha2)</p> <p>Devuelve el mayor de dos valores. Sólo puede ser utilizado para tipos de datos numéricos y fechas.</p> <p>Ejemplo:</p> <p>MAX(DTOC("01/01/2004"),DTOC("05/11/2010"))</p> <p>Resultado: 05/11/2010</p> <p>MAX(100,200)</p> <p>Resultado 200</p>
DTOC()	<p>Sintaxis: DTOC(dFecha)</p> <p>Devuelve en formato carácter el valor de un dato de tipo Fecha.</p> <p>Ejemplo:</p> <p>MsgAlert(DTOC(dDesde),"Fecha Desde")</p> <p>Resultado: '01/01/2004'</p>
ALLTRIM()	Sintaxis: ALLTRIM(cCadena)

	<p>Remueve los espacios en blanco de lado derecho e izquierdo de una cadena.</p> <p>Ejemplo ALLTRIM(STR(nSalario)) Resultado: '250000'</p>
LEFT()	<p>Sintaxis: LEFT(cCadena,nLen)</p> <p>Extrae desde el inicio de la Cadena la cantidad de caracteres solicitado por <nLen>.</p> <p>Ejemplo:</p> <p>LEFT('Programa',3)</p> <p>Resultado: 'Pro'</p>
LEN()	<p>Sintaxis: LEN(cCadena)</p> <p>Determina la longitud de una Cadena o de un Arreglo.</p> <p>Ejemplo:</p> <p>Len("N6mina")</p> <p>Resultado = 6</p> <p>Ejemplo:</p> <p>Len(ARRAY(20))</p> <p>Resultado 20</p>
RAT()	<p>Sintaxis: RAT(cChar,cCadena)</p> <p>Obtiene la posici6n desde la derecha que posee <cChar> dentro de <cCadena></p> <p>Ejemplo:</p> <p>RAT("N","NOMINA")</p> <p>Resultado 2</p>

<p>RIGHT()</p>	<p>Sintaxis: RIGHT(cCadena,nLen)</p> <p>Obtiene desde el final de la cadena la cantidad de caracteres indicados en <nLen></p> <p>Ejemplo:</p> <p>LEFT('Programa',3)</p> <p>Resultado: 'ama'</p>
<p>SUBS()</p>	<p>Sintaxis: SUBS(cCadena,nPos,nLen)</p> <p>Obtiene de derecha a izquierda y desde cualquier parte de la cadena especificado en <nPos> la cantidad de caracteres solicitados por <nChar></p> <p>Ejemplo:</p> <p>SUBS('Programa',2,3)</p> <p>Resultado: 'rog'</p>
<p>STRTRAN()</p>	<p>Sintaxis: STRTRAN(cCadena,cAnterior,cNuevo)</p> <p>Busca y reemplaza textos en una cadena, busca el valor de "cAnterior" y lo reemplaza por "cNuevo", realiza la lectura de derecha a izquierda.</p> <p>Ejemplo:</p> <p>STRTRAN('Programa','Pro','ADAPTAPRO')</p> <p>Resultado: 'ADAPTAPROgrama'</p>
<p>STR()</p>	<p>STR: (nValue,nEnteros,nDecimales)</p> <p>Convierte un valor numérico en expresión de tipo Carácter según la cantidad de enteros indicados en <nEnteros> y la cantidad de decimales indicados en</p>

	<p><nDecimales>. Los par3metros Enteros y Decimales pueden ser omitidos.</p> <p>Ejemplo:</p> <p style="text-align: center;">STR(1000,7,2)</p> <p>Resultado: '1000.00'</p>
--	--

OTRAS FUNCIONES

ASIGN_PRES()	<p>Sintaxis: ASIGN_PRES(FchInicio,FchFin)</p> <p>Obtiene el total de los conceptos (asignaciones y deducciones) que suman el salario integral dentro de un per3odo determinado.</p> <p>Ejemplo:</p> <p>//</p> <p style="text-align: center;">ASIGN_PRES("01/02/2000","29/02/2000")</p> <p>Resultado: Realiza la sumatoria de todos los conceptos que vayan al salario integral en per3odo correspondido entre el 01/02/2000 y el 29/02/2000</p>
ASIGN_NORM()	<p>Sintaxis: ASIGN_NORM(FchInicio,FchFin)</p> <p>Obtiene el total de los conceptos (asignaciones y deducciones) que suman el salario normal dentro de un per3odo determinado.</p> <p>Ejemplo:</p> <p>//</p> <p style="text-align: center;">ASIGN_NORM("01/02/2000","29/02/2000")</p> <p>Resultado: Realiza la sumatoria de todos los conceptos que vayan al salario normal en per3odo correspondido entre el 01/02/2000 y el 29/02/2000</p>
ACTUAL ()	<p>Sintaxis: ACTUAL (Cuota, Campo, Signo)</p> <p>Su funci3n es actualizar el incremento o decremento del Valor num3rico de un</p>

	<p>campo, seg3n otro valor determinado aplicando la siguiente ecuaci3n. $Campo=Campo+Cuota$ 3 $Campo=Campo-Cuota$.</p> <p>Ejemplo: //</p> <p>ACTUAL (500,"PRESTAMO",-")</p> <p>Resultado: Devuelve 500 y ejecuta en el campo: "PRESTAMO" La resta de 500 si el valor anterior era 1000 ahora ser3 500.</p>
<p>DIV()</p>	<p>Sintaxis: $DIV(nValor,nDiv)$</p> <p>Obtiene el valor de la divisi3n de <nValor> entre <nDiv> evitando el error de DIVIDE BY ZERO que pueda generar la divisi3n de un valor entre "0" cero.</p> <p>Ejemplo:</p> <p>$DIV(30000,10)$</p> <p>Resultado: 3000</p>
<p>CREAVAR()</p>	<p>Sintaxis: $CREAVAR(Variable,Contenido)$</p> <p>Permite crear variables para almacenar datos de cualquier tipo de datos.</p> <p>Ejemplo:</p> <p>//</p> <p>CREAVAR("FECHA",DATE())</p> <p>Crea la variable de nombre FECHA con el valor de la funci3n DATE()</p>
<p>FUNCION ()</p>	<p>Sintaxis: $FUNCION (nNumero)$</p> <p>Ejecuta la f3rmula del concepto definido como funci3n o de tipo "Fnnn" y devuelve el resultante de la misma. <nNumero> ser3 convertido en "F"+nNumero, construyendo "Fnnn" que identifica a la funci3n que ser3 ejecutada, en caso de no existir "Fnnn" ser3 desplegado un mensaje de advertencia.</p>

	<p>Ejemplo:</p> <p>Formula del concepto F001: Return 1+1</p> <p>Resultado: 2</p> <p>Formula del concepto A001: 2*FUNCION(1)</p> <p>Resultado: 4</p> <p>Formula del concepto A002: 4*FUNCION(1)</p> <p>Resultado: 8</p>
<p>LSTR()</p>	<p>Sintaxis: LSTR(nValor)</p> <p>Devuelve en formato de cadena, sin decimales y sin espacios el valor de una expresi3n num3rica.</p> <p>Ejemplo:</p> <p>VAROBSERV:="Salario :"+LSTR(SALARIO)</p>
<p>REDONDEA()</p>	<p>Sintaxis: REDONDEA (Monto)</p> <p>Permite redondear el monto suministrado a su valor mas alto o bajo en decimales respectivamente y dependiendo de la configuraci3n inicial de redondeo en NMCONFIG.</p> <p>Ejemplo:</p> <p>//</p> <p>REDONDEA (500.10)</p> <p>Resultado: Si la configuraci3n inicial de redondeo esta activada, la funci3n retornar3 500.</p>
<p>SUM()</p>	<p>Sintaxis: SUM (cCampo,cCondicion)</p> <p>Obtiene la suma de cualquier campo de tipo num3rico de la tabla trabajadores. El proceso recorre todos los registros y totaliza el campo solicitando y descartando los registros que no cumplan con la condici3n dada.</p> <p>Ejemplo:</p>

	<p>SUM("SALARIO","COD_DEP='ADM'")</p> <p>Resultado: Suma total del campo "SALARIO" de todos los trabajadores del departamento de administraci6n.</p>
JOR_SEMHRN()	<p>Sintaxis: JOR_SEMHRN (cJornada,@cTime)</p> <p>Obtiene la cantidad de horas semanales laborables registradas en cada c6digo de Jornada. <cJornada> se refiere al c6digo de Jornada. <@cTime> contiene en formato texto la cantidad de horas basadas en horas y minutos "hh:mm"</p> <p>Ejemplo:</p> <p>JOR_SEMHRN("S") // 40</p> <p>Resultado: Devuelve la cantidad de horas laborables registradas para la jornada semanal</p>
NTIME()	<p>Sintaxis: NTIME(cTime1,cTime2)</p> <p>Obtiene la cantidad de horas entre dos horarios, cada hora debe ser enviada en formato "HH:MM", horas y minutos m6s el sufijo "A" donde se refiere "Antes meridiano o "P" referido a "Post Meridiano"</p> <p>Ejemplo:</p> <p>NTIME("08:00A","12:00A") // 4</p> <p>Devuelve Cuatro (4) horas</p>
PORCEN()	<p>Sintaxis: PORCEN(nMonto,nTasa)</p> <p>Obtiene el valor porcentual indicado en <nTasa> y aplicado sobre el valor de <nMonto></p> <p>Ejemplo:</p> <p>PORCEN(SALARIO,20)</p> <p>Resultado: Obtiene el 20% del Salario.</p>

INFORME

LISTADOS DE CONCEPTOS DE PAGOS

Este reporte muestra todo los conceptos de pagos, ordenados por código de concepto, a través de los campos; código, descripción, tipo de nómina, salario que afecta (básico, integral, utilidad, vacación, alterado y fecha.

JORNADAS Y TURNOS

Muestra las jornadas y turnos definidos para el cálculo de horarios, está ordenado por código, a través de los campos; código y descripción.

FORMULAS DE LOS CONCEPTOS

Imprime o muestra las formulas de cada concepto, a través de los campos; código, descripción y formula. Está ordenado por código de concepto

CONSTANTES DE PAGO

Este reporte muestra todas las constantes, ordenadas por código de constante, a través de los campos; código, descripción y valor de la constante.

TASAS DE INTERÉS

Este reporte muestra todas las tasas de interés, ordenadas por fecha, a través de los campos; fecha, prestaciones, banca, otras tasas y préstamos.

DÍAS FERIADOS

Muestra todos los días feriados del año a través de los campos; Descripción, Día y Mes.

OTRAS NÓMINAS

Muestra las otras nóminas y está ordenado por código, a través de los campos; Código y Descripción.

CUENTAS CONTABLES

Muestra todas las cuentas contables y está ordenado por código, a través de los campos; Código y Descripción.

TIPOS DE AUSENCIA

Este reporte muestra las ausencias laborales a través de los campos; apellido y nombre del trabajador, número, fecha desde, fecha hasta, motivo y descripción. Está ordenado por el código del trabajador.

CARTAS DE TRABAJO

Este reporte muestra todas las cartas de trabajo, el mismo muestra los campos; Descripción y nombre del archivo.

UNIDADES TRIBUTARIAS

Muestra el valor de las unidades tributarias a través de los campos; fecha, monto, gaceta y fecha de la gaceta. Está ordenado por fecha de entrada en vigencia.

TARIFAS DE IMPUESTO SOBRE LA RENTA

Muestra todas las tarifas de Impuesto Sobre la Renta a través de los campos; U.T. Hasta, Porcentaje, Sustraendo UT. Y está ordenado por U.T. Hasta y Sustraendo UT.

PROCESO

IMPORTAR EJEMPLO

El ejemplo contenido en ADAPTAPRO NOMINA contiene los conceptos de pagos con sus respectivas formulas, constantes, etc. Ejecutando este proceso permite importar a la empresa deseada estos conceptos. Adicionalmente permite importar datos de trabajadores, departamento, etc. para realizar pruebas.

Al indicarle “Acepto” pasa las formulas y constantes, y conjuntamente con la opción “Importar datos para realizar pruebas”, pasa también los datos de los trabajadores.

IMPORTAR DATOS DESDE DPNM24

Permite importar las tablas y los recibos de pago desde Nómina 2.4., antes de realizar este proceso es necesario importar los conceptos (ejemplo) necesarios para la integridad de la tabla histórica “nmhistorico”.

El proceso lee los conceptos de nómina 2.4 faltantes en la tabla de conceptos y los incorpora bajo la condición inactivos y la fórmula traducida al procedimiento de conceptos de ADAPTAPRO NOMINA en forma comentada “/* */”. Los conceptos importados desde DpNm24 no generan ningún tipo de resultado.

Si la tabla de trabajadores tiene registros no será posible ejecutar el proceso de importación.

Primero inicia el proceso de importación de trabajadores y tablas relacionadas, luego recibos de pago con su respectivo detalle o histórico de pagos.

La tabla de fechas es generada en forma automática según los periodos importados.

Recomendación:

Antes de iniciar el proceso de importación, ejecute reorganizar archivos índices desde DPNM24 para asegurarse que los archivos de índices CDX sean confiables.

Si posee una versión menor a la 2.4, es necesario actualizarla a la 2.4, para luego hacer la importación a ADAPTAPRO NOMINA

EXPORTAR CONCEPTOS

Este proceso permite exportar o llevar los conceptos de pagos a otra ubicación en formato DBF. Esta opción es útil cuando se modifica un concepto y sólo se desea pasar ese.

IMPORTAR CONCEPTOS

Este proceso permite importar o traer los conceptos de pagos a ADAPTAPRO NOMINA. Esta opción es útil cuando se modifica un concepto y sólo se desea recibir ese.

OTROS

Está compuesto por un conjunto de programas diseñados para la configuración del sistema de Nómina.

SISTEMA ADMINISTRATIVO CONTABLE

Permite hacer la conexión con el Sistema Administrativo y Contabilidad con la finalidad de contabilizar y generar los asientos en Caja y Banco.

FICHA DEL TRABAJADOR

Permite agregar, modificar o eliminar campos al formulario del trabajador.

DENOMINACIÓN DE TÉRMINOS

Tiene como finalidad permitirle al usuario ajustar o definir los términos de las principales tablas de datos que dispone el sistema, logrando mayor comodidad en su utilización. Como por ejemplo el nombre de la tabla "Grupos" que puede ser empleada para: Agencias, Sucursales, Obras, etc. A través de esta opción es posible cambiar las denominaciones de las tablas: Grupos, Sucursales, Departamentos, Unidad Funcional y Cargos. Además también puede definir el nombre para los cuatro tipos de salarios "A,B,C y D" que se almacenan en la tabla "Resumen Mensual por Trabajador".

La nomenclatura utilizada por el sistema para asumir el nombre de las tablas se define a través de la siguiente regla. "oDp:NMTABLA" donde "oDp", se refiere al objeto de valores públicos o globales del sistema y "NMTABLA" se refiere al nombre específico de cualquier tabla. La función GETFROMVAR(), devuelve su contenido, como por ejemplo: GETFROMVAR("{oDp:NMGRUPO}") devuelve el nombre de la tabla "NMGRUPO" y el resultado puede ser: Grupo, Agencia, Sucursal, Obra, entre otros. Por tal motivo gran parte del diseño del sistema han sido creados bajo este esquema, entre ellos tenemos las opciones del Menú del Sistema, Formularios y Editores de Registros.

Igualmente la definición de valores por defecto para estas tablas es definida mediante este programa y tiene como función indicar los valores que debe ser asumido en caso de ser omitidos por el usuario en el momento de incluir un trabajador desde el formulario de trabajadores. Como por ejemplo, el valor que debe asumir la mayoría de trabajadores referenciado con la tabla "Grupos". En la mayoría de los casos se define un valor por defecto que debe estar registrado en la tabla "Grupos" y sugerido en la ficha del trabajador en el momento que se selecciona la opción "Incluir".

Para la asignación de valores de las tablas y tipos de costos el sistema presenta un formulario seleccionado en dos grupos de trabajo, el primero se refiere a los nombres y valores por defecto de cada tabla "Grupos, Departamentos, Unidad Funcional, Profesiones y Cargos" y el segundo grupo se refiere al nombre para los salarios "A,B,C y D"

Para la definición de la denominación de las tablas el formulario dispone de cuatro casillas por cada tabla y se tienen las siguientes funciones:

Nombre: Se refiere a la descripción de la tabla definido en el diccionario de Datos. Según este nombre se titularan las opciones del menú principal y editores de registros. Como por ejemplo, la tabla "Grupos" asume el nombre "Sucursales", las opciones del menú y editores de registros de grupos serán presentadas con el nuevo nombre "Sucursales".

Singular: De la misma forma que el nombre de la tabla, el valor de esta casilla se refiere al nombre singular de la tabla o valor que representa y tiene mayor utilización en los formularios de carga de datos,

como por ejemplo: Un trabajador está asociado a un “Grupo” y no a un “Grupos”, por razones semánticas es necesario que cada tabla también posea un nombre para denominar las situaciones donde se exige uno de sus registros. Si el nombre de la tabla “Grupos” es cambiado por “Sucursales” el formulario de trabajadores debe solicitar el código de “Sucursal”. La nomenclatura y sintaxis para obtener el nombre singular de la tabla desde el diccionario de datos es: `GETFROMVAR({oDp:xNMGRUPO})` devuelve: Grupo.

Códigos por Defecto: Permite introducir el valor del código por defecto que será asumido en el formulario de “Trabajadores” por cada tabla siempre que el usuario desee incluir un trabajador. Cada casilla está acompañada por un botón que muestra todos sus registros con la finalidad de facilitar la selección de un código que será importado hacia la casilla. El valor de esta casilla puede quedar vacío en caso que no sea requerido utilizar los códigos por defecto, en caso de no estar vacío el programa valida que el valor introducido esté registrado en su respectiva tabla y finalmente es mostrado su valor al lado izquierdo de la casilla. Los valores de cada código son almacenados en el DataSet por empresa y se identifican a través de la siguiente sintaxis:

Grupos: oDp:cGruSug
Unidad Funcional: oDp:cUndSug
Profesión: oDp:cPrfSug
Departamento: oDp:cDptSug
Cargos: oDp:cCarSug

Salarios: Presenta cuatro casillas “A, B, C y D” se refieren a la denominación que debe poseer cada tipo de Salario para la tabla “Resumen Mensual por Trabajador” o exigido en el formulario de definición de conceptos. Por defecto los valores son: Básico, Integral, Utilidades y Vacaciones.

Al finalizar la carga de datos en cada casilla, el programa presenta dos botones:

Aceptar: Inicia el proceso de actualización del `nmdataset` y Actualización del Diccionario de Datos.

Salir: No ejecuta ningún proceso y cierra el formulario.

CUENTAS DE NÓMINA POR PAGAR

Son utilizadas cuando la empresa decide llevar una cuenta contable general por tipo de nómina.

Cuenta	Nómina	Descripción de la Cuenta
Semanal:	5.2.4.01.12	SALARIOS DE OBREROS
Quincenal:	5.2.4.02.11	SUELDOS EMPLEADOS
Mensual:	5.2.4.02.10	SUELDOS DIRECTORES
Otra:	5.2.4.01.18	VACACIONES

Mayorías en el Sistema de Nómina:

08/11/2010

170 Mejoras en programa NMCONFIG, "Configurar Empresas", nueva pestaña "Salarios" que permite asignar valores en las nuevas Variables lógicas. permiten el cálculo de salarios promedios basados en dos esquemas, según (Acumulados) o (Histórico de Pagos) Los conceptos para los cálculos de Utilidades, Vacaciones, Antigüedad laboral, puede ser obtenidos desde el "Acumulado de Promedios" o desde el "histórico de pagos".

Para utilizar el "Histórico de Pagos" en el cálculo los salarios, debe activar

La casilla de salario deseado.

Según sea su selección, las opciones están contenidas en las siguientes variables:

En caso de seleccionar Histórico:

- oDp:lSalarioA:=.T.
- oDp:lSalarioB:=.T.
- oDp:lSalarioC:=.T.
- oDp:lSalarioD:=.T.

En caso de seleccionar Acumulados:

- oDp:lSalarioA:=.F.
- oDp:lSalarioB:=.F.
- oDp:lSalarioC:=.F.
- oDp:lSalarioD:=.F.

Estas variables, deben ser empleadas en los conceptos de pago, según el siguiente ejemplo:

Concepto A054 (Utilidades del Ejercicio)

```
IF oDp:lSalarioC
  // Basado en el Histórico
  nSalario:=ACUMC_FCH(oNm:aConceptosB,dDesde,oNm:dHasta)
  nSalario:=DIV(nSalario,nMeses)
ELSE
  // Basado en los Acumulados
  nSalario:=PROMEDIO_C(FCHINICIO,oNm:dHasta)
ENDIF
```

22/11/2010

171 Mejora en programa INTERESES, se agrego un nuevo parámetro cConIni "Concepto de Intereses", por defecto es N411, Registrados en la Carga inicial de datos. Logrando así, sumar los intereses cargados directamente más los Intereses calculados por este programa. Así mismo, el interés registrado en concepto N411, en la casilla siete "Intereses", fue truncado los terceros Decimales.

22/11/2010

173 Mejoras en programa NMPRESTVIEW, Visualizar Estado de Cuenta sobre Antigüedad laboral. Toma los Intereses del programa INTERESES y agrega los % de intereses como nuevas líneas en aquellos meses en donde no hay Antigüedad Laboral.

25/11/2010

172 Mejora en programa INTERESES, parametro "cAbonos", asume por defecto cAbonos:=cPagos:=oDp:cConAdel+", "+oDp:cConInter Anticipo de Antigüedad y Pago de Intereses Sobre antigüedad. Igualmente los programas NMCALINTERES (Intereses) y NMPRESTVIEW (Edo de cuenta), incluyen en su ejecucion el valor de exigido por este parámetro

30/11/2010

173 Mejora en programa NMPRESTVIEW, Estado de cuenta de prestaciones. Utiliza todos los datos utilizados por el programa intereses. Ahora se presenta con doce columnas.