

UNIVERSIDAD POLITECNICA DE VALENCIA

ESCUELA POLITECNICA SUPERIOR DE GANDIA

I.T. TELECOMUNICACIÓN (SIST. ELECTRONICOS)



UNIVERSIDAD
POLITECNICA
DE VALENCIA



ESCUELA POLITECNICA
SUPERIOR DE GANDIA

**“Desarrollo de un sistema para la
ejecución de procesos diferidos
basados en el paquete
DBMS_SCHEDULER de Oracle”**

TRABAJO FINAL DE CARRERA

Autor:
Efrén Martí Sarrió

Tutor:
Jordi Bataller Mascarell

GANDIA, 2014

Título : Desarrollo de un sistema para la ejecución de procesos diferidos basados en el paquete DBMS_SCHEDULER de Oracle.

Asunto : Memoria de Proyecto Fin de Carrera de Ingeniería Técnica en Telecomunicaciones

Autor : Efrén Martí Sarrió

Tutor : Jordi Bataller Mascarell

Universidad Politécnica de Valencia

Escuela Politécnica Superior de Gandia



Índice general

1	Introducción.....	13
2	Descripción de la tecnología.....	15
2.1	Oracle Database.....	15
2.2	PL/SQL.....	16
2.3	Oracle Developer Suite.....	17
	Oracle Forms.....	17
	Oracle Reports.....	17
	Oracle Jdeveloper.....	17
3	Análisis de requerimientos.....	19
3.1	Definición tareas programadas.....	19
3.2	Scheduler.....	20
	Vista General.....	20
	¿Que puede hacer Scheduler?.....	20
	Conceptos Básicos.....	21
	Programas.....	21
	Planificar horarios.....	22
	Trabajos o tareas.....	22
	Argumentos de las tareas.....	22
	Instancias de las tareas.....	23
	Eventos.....	23
	Procesos o cadenas.....	23
	Como están relacionados los programas, las tareas y las planificaciones.....	24
	Conceptos Avanzados.....	25
	Clases de tareas.....	25
	Ventanas de trabajo.....	25
	Grupo de ventanas de trabajo.....	26
	Arquitectura.....	26
	Tabla de tareas.....	27
	Coordinador de tareas.....	27
	Como se ejecutan las tareas.....	28
	Tareas esclavas.....	28
	Usando Scheduler en un entorno de aplicación real.....	28
	Como Funciona Scheduler cuando se tiene afinidad a un Servicio.....	29
3.3	Especificación Funcional.....	31
	Módulo General.....	32
	Mantenimiento Clases de Tareas.....	32
	Descripción Funcional.....	32
	Mantenimiento Ventanas de trabajo.....	34
	Descripción Funcional.....	34
	Mantenimiento Planificaciones.....	35
	Descripción Funcional.....	35
	Mantenimiento Eventos.....	37



Descripción Funcional.....	37
Módulo Gestión Programas.....	38
Mantenimiento Programas.....	38
Descripción Funcional.....	40
Módulo Gestión Cadenas.....	42
Mantenimiento Cadenas.....	42
Descripción Funcional.....	44
Módulo Gestión Tareas.....	47
Mantenimiento Tareas.....	47
Descripción Funcional.....	50
Módulo Gestión Suscripciones.....	53
Mantenimiento de Suscriptores.....	53
Descripción Funcional.....	56
Diseño Gráfico de la Aplicación.....	57
Mantenimiento Clases de Tarea.....	57
Mantenimiento Ventanas de Trabajo.....	58
Mantenimiento Planificaciones.....	59
Mantenimiento Eventos.....	62
Gestor de Programas.....	63
Gestor de Cadenas.....	65
Gestor de Tareas.....	68
Gestor de Suscriptores.....	71
4 Disección de la aplicación.....	73
4.1 Diagrama de Clases.....	73
Conceptos básicos.....	73
Elementos.....	73
Relaciones entre clases.....	74
Diagrama Clase Scheduler.....	75
4.2 Esquema Base de Datos.....	77
Modelo Entidad-Relación.....	77
Diagrama E-R del Modelo de Datos.....	78
5 Organización del código.....	79
5.1 Implementación Modelo.....	80
Introducción PL/SQL.....	80
Scheduler Común.....	82
Scheduler Windows.....	87
Scheduler Schedules.....	90
Scheduler Programs.....	92
Scheduler Chains.....	96
Scheduler Jobs.....	100
5.2 Implementación Vista – Controlador.....	106
Introducción Oracle Forms Developer.....	106
Barra de herramientas.....	107
Mantenimiento Clase de Tarea.....	108
Mantenimiento Ventanas de Trabajo.....	110



Mantenimiento de Planificaciones.....	111
Detalle Planificación.....	112
Mantenimiento de Eventos.....	113
Gestor de Programas.....	114
Gestor de Cadenas.....	117
Gestor de Tareas.....	121
Gestor de Suscripciones.....	124
6 Manual de instalación y usuario.....	127
6.1 Manual instalación de la aplicación.....	127
Crear Esquema Aplicación.....	127
Crear Directorio Aplicación.....	128
Configuración.....	128
6.2 Manual Usuario.....	129
Introducción.....	130
¿Qué se entiende por un proceso Batch?.....	130
Desde el punto de vista del usuario.....	130
Desde el punto de vista del sistema.....	130
Desde el punto de vista de la explotación/operación.....	130
Definición de un proceso Batch en SCHEDULER.....	130
Opciones de Administración.....	132
Definiciones.....	132
Clases de Tarea.....	132
Ventanas de Ejecución.....	132
Planificaciones.....	133
Eventos Externos.....	133
Diálogos y Pantallas.....	134
Mantenimiento de Clases de Tarea.....	134
Alta de Clases de Tarea.....	134
Modificación Clases de Tarea.....	135
Baja Clases de Tarea.....	135
Consulta de Ventanas de Ejecución.....	136
Mantenimiento Planificaciones.....	138
Alta de Planificación.....	139
Modificación Planificación.....	139
Baja Planificación.....	139
Evaluar Planificación.....	139
Mantenimiento de Eventos Externos.....	140
Alta de Evento.....	141
Modificación de Evento.....	141
Baja de Evento.....	141
Generar Evento.....	141
Gestión Programas.....	142
Definición de Programa.....	142
Programa.....	142
Parámetros.....	142



Resultados.....	142
Diálogos y Pantallas.....	143
Datos Generales.....	144
Alta de Programa.....	144
Modificación Programa.....	145
Baja Programa.....	145
Creación Proceso Simple.....	145
Mantenimiento de Parámetros.....	146
Alta de Parámetro.....	147
Modificación Parámetro.....	147
Baja de Parámetro.....	147
Mantenimiento de Resultados.....	148
Alta de Resultado.....	149
Modificación de Resultado.....	149
Baja de Resultado.....	149
Consulta de Cadenas y Tareas.....	150
Activar Programa.....	150
Desactivar Programa.....	150
Gestión de Cadenas.....	151
Definición de Cadenas.....	151
Cadena.....	151
Pasos y Dependencias.....	151
Definición de pasos.....	151
Definición de dependencias.....	152
Diálogos y Pantallas.....	154
Resumen Cadena.....	154
Activación de una Cadena.....	155
Des-activación de una Cadena.....	155
Copiar Cadena.....	155
Informe de Cadena.....	155
Mantenimiento de Cadenas (Datos Generales).....	156
Alta Cadena.....	157
Modificación Cadena.....	157
Baja Cadena.....	157
Establecer condición.....	157
Mantenimiento de Pasos y Dependencias.....	158
Alta de Paso.....	159
Modificación de Paso.....	159
Baja de Paso.....	159
Crear Pasos Secuenciales.....	159
Establecer Condición.....	159
Consulta de Tareas.....	160
Gestión de Tareas.....	161
Definición de Tareas.....	161
Planificación de Tareas.....	161



Estados de una Tarea.....	161
Diálogos y Pantallas.....	163
Gestión Tareas.....	163
Activación de una Tarea.....	164
Des-activación de una Tarea.....	164
Para una Tarea.....	164
Ejecución Inmediata de una Tarea.....	165
Consultar Programa/Cadena.....	165
Mantenimiento Tareas.....	166
Alta Tarea.....	167
Modificación Tarea.....	168
Baja Tarea.....	168
Establecer Planificación.....	168
Consultar Próximas Ejecuciones.....	168
Parámetros Tareas.....	169
Monitorizar Ejecución.....	170
Refrescar.....	171
Parar Tarea.....	171
Modificar Estado Paso.....	171
Suscripciones de una Tarea.....	172
Gestión de Suscripciones.....	173
Definición de Suscripción.....	173
Diálogos y Pantallas.....	173
Suscripciones por Tarea.....	174
Suscripciones por usuario.....	175
Alta Suscripción.....	177
Modificación Suscripción.....	177
Baja Suscripción.....	177
Activación Suscripción.....	177
Des-activación Suscripción.....	178
Personalizar.....	178
Usuario.....	178
7 Conclusiones y trabajo futuro.....	179
8 Bibliografía.....	181



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Trabajo Final de Carrera



Índice de ilustraciones

Ilustración 1: Relaciones entre los Programas, Tareas y Planificaciones.....	24
Ilustración 2: Ventana de trabajo para asignar recursos a las tareas.....	26
Ilustración 3: Componentes Scheduler.....	27
Ilustración 4: Arquitectura RAC en Scheduler.....	29
Ilustración 5: Scheduler, Afinidad a un Servicio.....	30
Ilustración 6: Estructura Aplicación DBMS_SCHEDULER.....	31
Ilustración 7: Estados de los Pasos de una Cadena.....	42
Ilustración 8: Estados de una Cadena.....	48
Ilustración 9: Modelo Comunicación Gestor Suscriptores.....	54
Ilustración 10: Formulario Mantenimiento Clases de Tarea.....	57
Ilustración 11: Formulario Mantenimiento Ventanas de Trabajo.....	58
Ilustración 12: Formulario Mantenimiento Planificaciones.....	59
Ilustración 13: Formulario Detalle Planificación.....	61
Ilustración 14: Formulario Mantenimiento Eventos.....	62
Ilustración 15: Formulario Gestor Programas (Datos Generales).....	63
Ilustración 16: Formulario Gestor Programas (Parámetros / Argumentos).....	63
Ilustración 17: Formulario Gestor Programas (Resultados).....	64
Ilustración 18: Formulario Gestor Cadenas (Gestión).....	65
Ilustración 19: Formulario Gestor Cadenas (Datos Generales / Condiciones Finalización).....	65
Ilustración 20: Formulario Gestor Cadenas (Pasos y Dependencias).....	66
Ilustración 21: Formulario Gestor Cadenas (Establecer Condición).....	67
Ilustración 22: Formulario Gestor Tareas (Gestión).....	68
Ilustración 23: Formulario Gestor Tareas (Datos Generales).....	68
Ilustración 24: Formulario Gestor Tareas (Parámetros).....	69
Ilustración 25: Formulario Gestor Tareas (Monitorizar Ejecución).....	69
Ilustración 26: Formulario Gestor Suscripciones (Suscripciones).....	71
Ilustración 27: Formulario Gestor Suscripciones (Todas).....	71
Ilustración 28: Diagrama Clases. Conceptos Básicos.....	73
Ilustración 29: Diagrama Clases. Scheduler.....	75
Ilustración 30: Diagrama ER. Scheduler.....	78
Ilustración 31: Modelo Vista Controlador (MVC).....	79
Ilustración 32: Scheduler Comun.....	82
Ilustración 33: Scheduler Windows.....	87
Ilustración 34: Scheduler Schedules.....	90
Ilustración 35: Scheduler Programs.....	92
Ilustración 36: Scheduler Chains.....	96
Ilustración 37: Scheduler Jobs.....	100
Ilustración 38: Barra de Herramientas.....	107
Ilustración 39: Mantenimiento Clase de Tarea.....	108
Ilustración 40: Mantenimiento Ventanas de Trabajo.....	110
Ilustración 41: Mantenimiento Planificaciones.....	111
Ilustración 42: Detalle Planificación.....	112



Ilustración 43: Mantenimiento de Eventos.....	113
Ilustración 44: Gestor Programas - Datos Generales.....	114
Ilustración 45: Gestor Programas - Parámetros.....	115
Ilustración 46: Gestor Programas - Resultados.....	115
Ilustración 47: Gestor Cadenas - Gestión.....	117
Ilustración 48: Gestor Cadenas - Datos Generales.....	117
Ilustración 49: Gestor Cadenas - Pasos y Dependencias.....	118
Ilustración 50: Gestor Cadenas - Establecer Condición.....	119
Ilustración 51: Gestor Tareas - Gestión.....	121
Ilustración 52: Gestor Tareas - Datos Generales.....	122
Ilustración 53: Gestor Tareas - Parámetros.....	122
Ilustración 54: Gestor Tareas - Monitorizar Ejecución.....	123
Ilustración 55: Gestor de Suscriptores - Suscripciones.....	124
Ilustración 56: Gestor de Suscripciones - Todas.....	125
Ilustración 57: Manual Usuario. Clases de Tarea.....	132
Ilustración 58: Manual Usuario. Ventanas de Ejecución.....	132
Ilustración 59: Manual Usuario. Planificaciones.....	133
Ilustración 60: Manual Usuario. Eventos Externos.....	133
Ilustración 61: Manual Usuario. Mantenimiento Clases de Tarea.....	134
Ilustración 62: Manual Usuario. Consulta Ventanas de Ejecución.....	136
Ilustración 63: Manual Usuario. Mantenimiento Planificaciones.....	138
Ilustración 64: Manual Usuario. Detalle Planificación.....	138
Ilustración 65: Manual Usuario. Mantenimiento Eventos Externos.....	140
Ilustración 66: Manual Usuario. Gestión Programas.....	143
Ilustración 67: Manual Usuario. Gestión Programas. Parámetros.....	146
Ilustración 68: Manual Usuario. Gestión Programas. Resultados.....	148
Ilustración 69: Manual Usuario. Gestión Programas. Consulta Cadenas y Tareas.....	150
Ilustración 70: Manual Usuario. Gestión Cadenas. Estados.....	151
Ilustración 71: Manual Usuario. Gestión Cadenas. Resumen.....	154
Ilustración 72: Manual Usuario. Gestión Cadenas. Datos Generales.....	156
Ilustración 73: Manual Usuario. Gestión Cadenas. Condición Finalización.....	157
Ilustración 74: Manual Usuario. Gestión Cadenas. Pasos y Dependencias.....	158
Ilustración 75: Manual Usuario. Gestión Cadenas. Condición Ejecución.....	160
Ilustración 76: Manual Usuario. Gestión Cadenas. Consulta Tareas.....	160
Ilustración 77: Manual Usuarios. Gestión Tareas. Catálogo.....	164
Ilustración 78: Manual Usuario. Gestión Tareas. Datos Generales.....	166
Ilustración 79: Manual Usuario. Gestión Tareas. Parámetros.....	169
Ilustración 80: Manual Usuario. Gestión Tareas. Monitorizar Ejecución.....	170
Ilustración 81: Manual Usuario. Gestión Tareas. Suscripciones.....	172
Ilustración 82: Manual Usuarios. Gestor Suscriptores. Suscripciones por Tarea.....	174
Ilustración 83: Manual Usuario. Gestor Suscripciones. Suscripciones por Usuario.....	175
Ilustración 84: Manual Usuario. Gestor Suscripciones. Suscripciones por Usuario. Todas.....	175
Ilustración 85: Manual Usuario. Gestión Suscripciones. Usuarios.....	178



Índice de tablas

Tabla 1: Datos Clase de Tarea.....	32
Tabla 2: Datos Ventana de Trabajo.....	34
Tabla 3: Datos Planificaciones.....	35
Tabla 4: Datos Eventos Externos.....	37
Tabla 5: Datos Programas.....	39
Tabla 6: Datos Parámetros Programa.....	39
Tabla 7: Datos Resultados de Programa.....	39
Tabla 8: Estados de los Pasos de una Cadena.....	43
Tabla 9: Datos Cadena.....	44
Tabla 10: Datos Pasos de una Cadena.....	44
Tabla 11: Estados de las Tareas.....	48
Tabla 12: Datos Tarea.....	50
Tabla 13: Eventos Desencadenantes Envío Notificación.....	54
Tabla 14: Datos Suscriptor.....	55
Tabla 15: Datos Fichero Suscripciones.....	55
Tabla 16: Clausulas para definir el Intervalo de Repetición.....	60
Tabla 17: Estructura Procedimiento Almacenado.....	81
Tabla 18: Estructura Función.....	81
Tabla 19: Estructura Paquete.....	81
Tabla 20: pck_scheduler_comun.....	82
Tabla 21: atributos ventanas (windows attribute).....	84
Tabla 22: atributos planificaciones (schedule attribute).....	84
Tabla 23: atributos programa (program attribute).....	84
Tabla 24: atributos cadenas (chain attribute).....	85
Tabla 25: atributos clase de tareas (job class attribute).....	85
Tabla 26: atributos suscripciones (suscriber attribute).....	85
Tabla 27: atributos tarea (job attribute).....	86
Tabla 28: pck_scheduler_windows.....	87
Tabla 29: pck_scheduler_windows_group.....	89
Tabla 30: pck_scheduler_schedule.....	90
Tabla 31: atributos de scheduler (scheduler attribute).....	91
Tabla 32: pck_scheduler_program.....	92
Tabla 33: pck_scheduler_chains.....	96
Tabla 34: pck_scheduler_job_classes.....	100
Tabla 35: pck_scheduler_jobs.....	101
Tabla 36: Manual usuario. Clases de Tarea.....	134
Tabla 37: Manual Usuario. Consulta Ventanas Ejecución.....	137
Tabla 38: Manual Usuario. PPlanificación.....	139
Tabla 39: Manual Usuario. Eventos Externos.....	140
Tabla 40: Manual Usuario. Eventos Externos. Componentes.....	141
Tabla 41: Manual Usuario. Gestión Programas. Datos Generales.....	144
Tabla 42: Manual Usuario. Gestión Programas. Parámetros.....	146



Tabla 43: Manual Usuario. Gestión Programas. Resultados.....	148
Tabla 44: Manual Usuario. Gestión Cadenas. Estados.....	152
Tabla 45: Manual Usuario. Gestión Cadenas. Datos Generales.....	156
Tabla 46: Manual Usuario. Gestión Cadenas. Pasos y Dependencias.....	158
Tabla 47: Manual Usuario. Gestión Cadenas. Parámetros Pasos.....	159
Tabla 48: Manual Usuario. Gestión Tareas. Catálogo Tareas.....	163
Tabla 49: Manual Usuario. Gestión Tareas. Datos Generales.....	167
Tabla 50: Manual Usuario. Gestión Tareas. Parámetros.....	169
Tabla 51: Manual Usuario. Gestión Tareas. Monitorizar Ejecución.....	170
Tabla 52: Manual Usuario. Gestor Suscripciones. Datos Suscripción.....	176
Tabla 53: Manual Usuario. Gestor Suscripciones. Ficheros Suscripción.....	176



1 Introducción

La **Administración Pública** de la **Generalitat Valenciana** esta compuesto por un conjunto de *Organismos Públicos* que realizan la función administrativa y de gestión de la *Comunidad Valenciana* y otros entes públicos con personalidad jurídica, ya sean de ámbito regional o local.

La **Administración Electrónica** (*e-Administración*) hace referencia a la incorporación de las tecnologías de la información y las comunicaciones en dos vértices: desde un punto de vista intraorganizativo transformando las oficinas tradicionales en procesos electrónicos con el fin de crear una oficina sin papeles; desde una perspectiva de las relaciones externas habilitando la vía electrónica como un nuevo medio para la relación con el ciudadano y empresas. Es una herramienta con un elevado potencial de mejora de la productividad y simplificación de los diferentes procesos del día a día que se dan en los diferentes Organismos.

Ambos tipos de Administración utilizan la misma aplicación de gestión basado en un sistema **Oracle**. La ejecución de los diferentes Trámites de Gestión se realizan mediante procesos. La mayoría de estos procesos que se ejecutan son repetitivos y tediosos, los procesos que manejan grandes conjuntos de información son propensos a errores, si se realiza de forma manual, y precisan de un alto nivel de recursos por parte del Servidor. Este tipo de procesos se ejecutan de forma diferida (modo **batch**), es decir, sin la necesidad de interacción con el usuario.

Para crear procesos diferidos Oracle ofrece una funcionalidad de **Jobs** (*Tareas*). Dicha funcionalidad de Oracle es parecida al **Cron de Unix** en la cual se puede planificar una Tarea a una determinada hora y con una periodicidad concreta. La diferencia notable que se puede tener en cuenta entre el *Cron de Unix* y los *Jobs de Oracle* es obvia. Los *Jobs de Oracle* se programan dentro de la Base de Datos por lo tanto si la Base de Datos no está funcionando, los *Jobs* no se ejecutarán.

La planificación de *Jobs* se lleva al cabo a través del paquete **DBMS_JOB**. Dada la diferencia anotada anteriormente entre *Cron* y *Jobs* se concluye que la utilización de *Jobs* es conveniente utilizarlo cuando la Tarea que se realice afecte a la Base de Datos, que es la que tiene que estar funcionando. Para tareas de Sistema en este caso es más propio utilizar el *Cron* o "*Gestor de tareas programadas*" que tenga el Sistema Operativo.

El paquete **DBMS_JOBS** esta en desuso, y si existe es por cuestiones de compatibilidad de sistemas . *Oracle Corporation* recomienda fuertemente la migración de toda la programación de **DBMS_JOBS** a **DBMS_SCHEDULER**. Este nuevo paquete ofrece varias ventajas que su predecesor no ofrecía:

- Permite definir intervalos de ejecución de forma más expresiva y simple.
- Permite da nombres a las Tareas.
- Permite prioridades de ejecución.
- Permite Crear Ventanas de Ejecución con planes de recurso.
- Permite el almacenamiento de registros históricos (ejecución, errores, detalles de errores, etc.)
- Permite detener Tareas de forma más sencilla.

Objetivos del Proyecto

La creación, modificación, eliminación y consulta de Tareas diferidas recae sobre los *Administradores de Sistemas* mediante petición de los *Administradores de Aplicaciones*. Los *Administradores de Sistemas* se dedican exclusivamente al mantenimiento del sistema y de la Base de Datos, es decir, no tienen conocimiento funcional de la aplicación de Gestión. De echo el principal inconveniente de esta forma de trabajo es el número de problemas derivados de la mal interpretación de los



documentos mediante los que un *Administrador de Aplicación* encarga un trabajo a un *Administrador de Sistemas*, dándose el caso que este último tiene pocos o nulos conocimientos sobre la aplicación y el proceso que debe desarrollar. En línea con esto, el mantenimiento de estas Tareas diferidos resulta igualmente costoso e ineficiente que lo realice el *Administrador de Sistemas*.

Gracias a las ventajas que ofrece el paquete *DBMS_SCHEDULER* y por la necesidad de que el *Desarrollador de Aplicaciones* tenga control sobre las Tareas, se decide crear una Interfaz Gráfica de Usuario que permita Crear, Modificar, Eliminar y Consultar las Tareas sin la necesidad de realizar peticiones a los *Administradores de Sistemas* para realizar cualquier acción sobre la Tarea. Además se desarrollará un módulo que permita definir los resultados obtenidos tras la ejecución de la Tarea, y otro módulo que posibilita la suscripción a dichos resultados, bien por e-mail o por sms.

Por tanto el **objetivo de este proyecto** es la creación de una Interfaz Gráfica, utilizando las herramientas de Oracle, que facilite la creación de Tareas planificadas haciendo uso del paquete *DBMS_SCHEDULER*. En particular, gracias a este proyecto, un *Administrador de Aplicación*, podrá definir Tareas diferidas como por ejemplo el envío de notificaciones a gran escala. Un caso típico es enviar actualizaciones a todos los ayuntamientos y organismos públicos de la Comunidad Valenciana. Esta tarea debe realizarse normalmente en horas fuera de horario laboral para no colapsar el servidor. El script necesario para esto evidentemente necesita el conocimiento de las base de datos que contienen la información relevante y de las normativas aplicables. Actualmente el *Administrador de Sistemas* es ajeno a estos conocimientos pero es él el que debe escribir el script. Con la aplicación que se propone este script lo podrá realizar directamente un *Administrador de Aplicación*, lo cual agiliza el proceso y reducirá la cantidad de errores debidos a malentendidos, facilitando también su mantenimiento.

Organización de este documento

Este documento esta dividido en los siguientes puntos:

- Descripción de la tecnología:

En este punto se presentará, sin entrar en detalle, las herramientas utilizadas por la *Administración Pública* y que son necesarias para el desarrollo de la aplicación.

- Análisis de Requerimientos:

Se analiza de forma más detallada los requerimientos necesarios para el desarrollo de la Aplicación realizando un análisis funcional de como debería ser dicha aplicación.

- Disección de la aplicación:

Se utiliza un lenguaje de modelado gráfico para visualizar, especificar y documentar las partes que componen la aplicación.

- Organización del Código:

Utilizando un patrón de arquitectura de software se separa los datos y la lógica de negocio de la interfaz de usuario y el módulo que gestiona los eventos y comunicaciones.

- Manual de Instalación y Usuario:

Se expone como Instalar la aplicación en el entorno de la *Administración Pública* y se explica el uso de la aplicación.

2 Descripción de la tecnología

2.1 Oracle Database

Oracle database es un sistema de gestión de base de datos Objeto-Relacional (u ORDBMS por el acrónimo en inglés de *Object-Relational Data Base Management System*), desarrollado por Oracle Corporation.

Se considera a Oracle Database como uno de los sistemas de base de datos más completos, destacando:

- Soporte de transacciones
- Estabilidad
- Escalabilidad
- Soporte Multiplataforma

Soporte de transacciones es la propiedad que permite mantener la integridad de los datos haciendo que las transacciones no finalicen en un estado intermedio. Es decir, cuando por alguna causa el sistema debe cancelar la transacción, empieza a deshacer las órdenes ejecutadas hasta dejar la base de datos en su estado inicial, como si la transacción nunca se hubiese ejecutado.

Una **transacción** en un Sistema de Gestión de Base de Datos (SGBD), es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica. Las transacciones deben cumplir cuatro propiedades **ACID**:

1. **Atomicidad (Atomicity)**: propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedarse a medias.
2. **Consistencia (Consistency)**: propiedad que asegura que sólo se va a ejecutar aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.
3. **Aislamiento (Isolation)**: propiedad que asegura que una operación no puede afectar a otras.
4. **Permanencia (Durability)**: propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

La **estabilidad** es la propiedad de los sistemas que tienen un nivel de fallos reducido. A menos fallos, mayor estabilidad, y viceversa.

La **escalabilidad** es la propiedad que permite manejar grandes volúmenes de datos de forma fluida sin que los demás servicios se vean afectados.

Soporte Multiplataforma es la propiedad que permite ejecutar o interoperar en múltiples plataformas de desarrollo.

El dominio de Oracle en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia de *Microsoft SQL Server* y de la oferta de otros RDBMS con licencia libre como *PostgreSQL*, *MySQL* o *Firebird*.

Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo *GNU/Linux*.



2.2 PL/SQL

El **lenguaje de consulta estructurado** o **SQL** (por sus siglas en ingles *Structured Query Language*) es un lenguaje declarativo de acceso a base de datos relacionales que permiten especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permite efectuar consultas con el fin de recuperar de forma sencilla información de interés de la base de datos, así como hacer cambios en ella.

Para que un gestor de base de datos sea considerado como relacional, debe soportar SQL, independientemente de las características particulares que dicho gestor pueda aportar. Conocer SQL es conocer las bases de datos y todo su potencial.

SQL no posee la potencia de los lenguajes de programación, no permite el uso de variables, estructuras de control de flujo, bucles... y demás elementos característicos de la programación. No es de extrañar, SQL es un lenguaje de consulta, no un lenguaje de programación.

PL/SQL (*Procedural Language / SQL*) es el lenguaje de programación que proporciona Oracle para extender el SQL estándar con otro tipo de instrucciones y elementos propios de los lenguajes de programación. Con PL/SQL se programan las unidades de programa de la base de datos de Oracle, estas son:

- Procedimientos almacenados
- Funciones
- Triggers
- Scripts

El lenguaje PL/SQL esta incorporado en:

- Servidor de base de datos
- Herramientas de Oracle (Forms, Reports ...)

Casi todos los tipos de datos manejados por PL/SQL son similares a los soportados por SQL. Sin embargo cabe destacar que PL/SQL no es CASE-SENSITIVE, es decir, no diferencia mayúsculas de minúsculas como otros lenguajes de programación como C o Java. En cambio en la búsqueda de datos es CASE-SENSITIVE, es decir, si diferencia entre mayúsculas y minúsculas.

Para aprender más sobre PL/SQL se recomienda los siguientes documentos, incluidos en la bibliografía:

- Oracle Database, PL/SQL Packages and Types References, 10g Release 2 (10.2)
- Oracle JDeveloper 10g for Forms & PL/SQL Developers : A Guide to Web Development with Oracle ADF



2.3 Oracle Developer Suite

Oracle Developer Suite es un conjunto de herramientas de desarrollo publicados por la corporación Oracle. Inicialmente las herramientas que incorporaba eran *Oracle Forms* y *Oracle Reports*, posteriormente se introdujo *JDeveloper* y otras herramientas.

Oracle Forms

Oracle Forms es un software de desarrollo para la creación de formularios que interactúan con una base de datos de Oracle. Cuenta con un IDE que incluye un navegador, hoja de propiedades para los objetos y el editor de código que utiliza PL/SQL. Las versiones más recientes soportan Java y ejecutan sus formularios en un contenedor Java (Java EE), por tanto queda perfectamente integrado en Java y Servicios Web.

El enfoque principal de los formularios es la creación de sistemas de entrada de datos que acceden a una base de datos de Oracle.

¿Como Funciona?

Oracle Forms accede a la base de datos de Oracle y genera un formulario que presenta los datos. El formato fuente (*.FMB) se compila en un ejecutable (*.FMX), que se ejecuta (interpretando) por el módulo de tiempo de ejecución de los Formularios. El formulario se utiliza para ver y editar datos en aplicaciones con base de datos. Varios elementos de la GUI, como botones, menús, barras de desplazamiento, y gráficos se pueden colocar en el formulario.

Oracle Reports

Oracle Reports es un software para la elaboración de informes con los datos almacenados en una base de datos Oracle.

Los formatos de salida del informe pueden ser enviados directamente a una impresora o ser guardados en uno de los siguientes formatos:

- HTML
- RTF
- PDF
- XML
- Hoja de Cálculo

Oracle Jdeveloper

Oracle Jdeveloper es un software gratuito que suministra Oracle. Ofrece funciones para desarrollar en Java, XML, SQL i PL/SQL, HTML, JavaScript, BPEI i PHP. Jdeveloper cubre el ciclo de vida de un desarrollo completo, desde el diseño a través de la codificación, depuración, optimización y creación de perfiles.

Con Jdeveloper, Oracle a tratado de simplificar el desarrollo de aplicaciones, centrándose en proporcionar un enfoque visual y declarativo para el desarrollo de aplicaciones, además de la construcción de un entorno de codificación avanzado.





3 Análisis de requerimientos

3.1 Definición tareas programadas

Las **tareas programadas** son una lista de acciones que el Sistema Operativo debe ejecutar en segundo plano cada cierto tiempo. El periodo en el que se ejecuta cada una de las acciones puede variar entre:

- Diariamente
- Semanalmente
- Mensualmente
- Sólo una vez
- Al iniciar el equipo
- Al iniciar la sesión

En los sistemas operativos *Unix*, ***cron*** es el administrador que regula esta lista de acciones. Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el fichero *crontab*.

En los sistemas operativos *Windows*, es el **Gestor de tareas programadas** o **Administrador de tareas** el que se encarga de administrar los procesos y servicios que esta ejecutando el equipo en el momento.

En **Oracle Database** es **Scheduler** quien permite gestionar y programar las ejecuciones.

3.2 Scheduler

DBMS_SCHEDULER permite programar la ejecución de *bloques PLSQL*, así como *procedimientos y funciones PL/SQL*. Por otro lado, también permite programar la ejecución de binarios y *shell-scripts*.

DBMS_SCHEDULER es el paquete PLSQL que reemplazó en la versión de la base de datos Oracle 10g al paquete DBMS_JOB. Aunque el paquete DBMS_JOB sigue existiendo por razones de compatibilidad, no debe utilizarse ya que es muy probable que deje de existir en futuras versiones de las base de datos de Oracle.

Se va a realizar una introducción a los conceptos de Scheduler que incluyen los siguientes temas:

- Vista general
- Conceptos básicos
- Conceptos Avanzados
- Arquitectura

Para obtener más información sobre este paquete se puede consultar los capítulos 26, 27 y 28 del documento 1 que se menciona en la Bibliografía **Oracle Database 10g – Administrators guide 10.2**.

Vista General

Oracle ofrece un conjunto de procedimientos y funciones en el paquete DBMS_SCHEDULER para facilitar la gestión de tareas programadas, además de ofrecer un amplio conjunto de funcionalidades para las necesidades de programación de tareas más complejas. El conjunto de todas estas funcionalidades se llama **Scheduler**.

Scheduler permite a los administradores de base de datos y a los desarrolladores de aplicaciones controlar cuando se van a llevar al cabo las diversas tareas. Estas tareas pueden durar mucho tiempo, así que usar Scheduler puede ayudar a mejorar la gestión y la planificación. Además, asegurará que muchas de las tareas rutinarias se ejecutan sin intervención humana, reduciendo los costes operativos, si se implementan rutinas más fiables, se reduce al mínimo el error humano, y acortar las ventanas de tiempo necesarios.

Algunos ejemplos tipos donde se utiliza Scheduler son:

- Los administradores de base de datos pueden programar y supervisar las tareas de mantenimiento de base de datos recurrentes, tales como copias de seguridad o cargas nocturnas de almacenamiento de datos o extractos.
- Los desarrolladores de aplicaciones pueden crear programas o librerías de programas que los usuarios finales pueden utilizar en sus programas. Pudiendo controlar y supervisar la ejecución de estos programas.

¿Que puede hacer Scheduler?

Scheduler proporciona una serie de funcionalidades complejas que se pueden utilizar para:

- Ejecución de tareas planificados en tiempo o eventos.

Es la capacidad más básica de un gestor de tareas, dar la posibilidad de programar una tarea para ejecutarse a una determinada fecha y hora, o cuando se produce un evento en particular.

- Planificar las tareas según las necesidades

Cuando los recursos son limitados Scheduler reasigna las planificaciones de las tareas,



alineando los procesamientos de las tareas según las necesidades. Esto se lleva al cabo de las siguientes maneras:

- Las tareas que comparten características y comportamientos comunes se pueden agrupar en entidades más grandes llamadas clases de tarea. Asignando los recursos necesarios a la clase se pueden dar prioridades. Esto te asegura que las tareas críticas tienen prioridad y recursos suficientes para ejecutarse.
- Scheduler permite cambiar la asignación de prioridades en base a un horario. Debido a que una tarea puede convertirse en crítica en un determinado intervalo de tiempo.
- Además de planificar tareas basadas en horarios, se puede planificar tareas basadas en un sistema de eventos. Los programas pueden generar eventos que Scheduler puede reconocer. Según el evento Scheduler ejecutara la tarea asociada a dicho evento.
- Administrar y supervisar tareas
Las tareas sufren cambios de estado desde su inicio hasta el final de la ejecución. Esta actividad queda registrada y proporciona información valiosa acerca de las tareas y sus ejecuciones ayudando a programar y administrar mejor las tareas.
- Ejecutar y gestionar tareas en un entorno agrupado
Un *Cluster* es un conjunto de instancias de base de datos que coopera para realizar la misma tarea. *Oracle Real Application Cluster (RAC)* ofrece escalabilidad y fiabilidad sin ningún cambio en las aplicaciones. Scheduler es compatible con la ejecución de tareas agrupadas. Para equilibrar la carga en el sistema y para un mejor rendimiento, también se puede especificar el servicio de base de datos en la que la tarea debe ser ejecutada.

Conceptos Básicos

Scheduler ofrece un tratamiento modular para la gestión de tareas en el entorno de *Oracle*. Las ventajas de esta modalidad incluye una gestión más sencilla del entorno de base de datos y la reutilización de planificaciones para crear nuevas tareas similares a otras ya existentes.

Los elementos *Scheduler* se almacenan en tablas de base de datos, permitiendo utilizar los privilegios normales de *Oracle*.

Los elementos básicos de Scheduler son:

- **Programs** (Programas)
- **Schedules** (Planificar horarios)
- **Jobs** (Trabajos o tareas)
- **Events** (Eventos)
- **Chains** (Procesos o cadenas)

Programas

El elemento Programa, **Program Scheduler**, es una colección de metadatos acerca de lo que se va a ser ejecutado por el Scheduler. Incluye información como el nombre del programa, la acción (por ejemplo, un procedimiento o un ejecutable), el tipo de programa (por ejemplo, PL/SQL y procedimientos almacenados de java o procedimientos PL/SQL) y el número de argumentos requerido por el programa.

El programa esta separado de la tarea. Una tarea se ejecuta en un momento determinado, o porque se ha producido un evento determinado, e invocan al programa asignado. A las tareas se les asigna un



programa, de esta forma un programa puede ser asignado a diferentes tareas y ser ejecutado en diferentes momentos y con configuraciones diferentes. Teniendo en cuenta los privilegios adecuados, diferentes usuarios pueden utilizar el mismo programa sin tener que redefinirlo. Esto permite la creación de bibliotecas de programas, en el que los usuarios pueden seleccionar de una lista los programas existentes.

Debido a que un programa puede invocar a un procedimiento almacenado o ejecutables que requieran argumentos, se proporciona un medio para almacenar los valores predeterminados para estos argumentos.

Planificar horarios

El planificador de horarios, **Scheduler Scheduler**, especifica cuando y cuantas veces se ha de ejecutar una tarea. Las tareas pueden ser ejecutadas de forma inmediata o ser planificadas para que se ejecuten en un determinado momento. Para las tareas que se ejecuten en un determinado momento se puede especifica la fecha y hora en que la tarea se debe comenzar. Para las tareas que se repiten en un periodo de tiempo, se puede especificar una fecha y hora en que la tarea debe finalizar, lo que se conoce como caducidad.

Una tarea también se puede especificar para que se ejecute cuando se produce un determinado evento. Se hablará más de los eventos más adelante.

Al igual que los programas, las planificaciones se nombran y se almacenan en tablas de base de datos. Permitiendo compartir las planificaciones. Por ejemplo el final horario final de trabajo puede ser común para muchas tareas. En lugar de tener que definir la planificación cada vez que se cree la tarea, se puede asignar la planificación ya existente.

Algunos ejemplos de planificaciones que se pueden utilizar en tareas que se ejecutan en el tiempo son:

- Ejecuta el Miércoles, 25 de Diciembre, 2013 a las 14:00.
- Ejecutar todos los lunes, a las 8:00, a partir del 1 de Diciembre del 2013 hasta el 31 de diciembre del 2014.
- Ejecutar todos los días laborables.

Trabajos o tareas

Una tarea, **Job Scheduler**, es el elemento que se ejecuta. Es una combinación de lo que debe ser ejecutado (acción) y cuando (la planificación). Se pueden crear las siguientes tareas:

- Especificar los atributos de una tarea como la acción a realizar (por ejemplo, un bloque PL/SQL) y la planificación en que se debe ejecutar la acción (por ejemplo, todos los días al mediodía, o cuando se produce un determinado evento).
- Especificar los atributos de una tarea asignando un programa y planificación ya existentes.

Al igual que los programas y las planificaciones las tareas se nombran y se almacenan en tablas de base de datos.

Argumentos de las tareas

Cuando una tarea llama a un programa con argumentos, se pueden especificar el valor de estos argumentos en la tarea, prevaleciendo sobre los valores que se han creado en la definición del programa.

Una tarea no se puede habilitar si no se han definido todos los valores de sus argumentos ya sea a nivel de tarea o de programa.

Un ejemplo común son las tareas que ejecutan un conjunto de informes todas las noches. Es decir,

distintos departamentos de una empresa requieren diferentes informes. Para ello se puede crear un programa que ejecute un Script de Informes atendiendo a un argumento de entrada que será el número de departamento. Cada departamento creará su tarea que apuntará al programa, y especificará como argumento de la tarea el número de departamento.

Instancias de las tareas

Las instancias representan las ejecuciones específicas de una tarea. Las tareas que están planificadas para que se ejecuten una sola vez, tendrán una sola instancia. Las tareas que tienen un intervalo de repetición, tendrán varias instancias. Por ejemplo, una tarea que se ejecute el Primer Martes de Enero de 2014 tendrá una sola instancia. Una tarea que se ejecute todos los días al medio día durante una semana, tendrá siete instancias.

Cuando se crea una tarea, se añade en la tabla de tareas de Scheduler. Cada vez que se ejecute la tarea, se agrega una entrada en el registro de tareas. Por tanto, si se crea una tarea que se ejecuta repetidas veces, se encontrará una instancia en la tablas de tareas y varias instancias de esta tarea en la tabla de registros. Cada instancia de la tabla de registros proporciona información acerca de la ejecución como el estado al finalizar de la ejecución, fecha inicio y fin entre otros.

Eventos

Un evento, **Event Scheduler**, es un mensaje que envía una aplicación o proceso del sistema a otro para indicar que una acción o ocurrencia ha sido detectada. Se provoca un evento por aplicación o proceso (Enviado) y se consume por una o mas aplicaciones o procesos (Recibido).

Existen dos tipos de eventos en *Scheduler*:

- Eventos producidos por el propio *Scheduler*.

Scheduler genera eventos para indicar los cambios de estado que se producen dentro del mismo *Scheduler*. Por ejemplo, *Scheduler* genera un evento cuando se inicia la tarea, cuando se completa la tarea, cuando la tarea supera el tiempo de ejecución asignado, y así sucesivamente. El consumidor de estos Eventos puede ser otra tarea que realiza alguna acción en respuesta a dicho evento.

- Evento producidos por una Aplicación.

Una aplicación puede generar un evento para que sea consumida por *Scheduler*. Este reacciona al evento ejecutando una tarea. Se puede crear una planificación que hace referencia a un evento en vez de una franja horaria. Si la tarea esta asignada a dicha planificación, esta se ejecutara cuando se produzca el evento. También se pueden crear tareas sin asignar ninguna planificación haciendo una referencia directa al evento.

Scheduler utiliza *Oracle Streams Advanced Queuing* para enviar y recibir eventos. El procedimiento a seguir es el siguiente, desde *Scheduler* se crea un evento que esta asociado a una *queue* predefinida. Las aplicaciones se suscriben a dicha *queue* para incluir los mensajes. Al incluir un mensaje en la *queue* se generará un evento. *Scheduler* detecta dicho evento y ejecuta la tarea asociada. Esta tarea podrá leer el mensaje que la aplicación a incluido en la *queue*.

Para más información sobre las *Queue* se recomienda ver la siguiente documentación:

- *Oracle Streams, Advanced Queuing User's guide and Reference*

Procesos o cadenas

Una Cadena, **Chains Scheduler**, es un conjunto de programas que están unidos entre si para lograr un objetivo. Un ejemplo de una cadena puede ser "Ejecutar el programa A. Ejecutar el Programa B. Ejecutar el Programa C si el programa A y el B se han completado con éxito de lo contrario ejecutar el programa D". A las tareas se les puede asignar una Cadena en vez de un programa.

Cada posición de una Cadena que ejecuta un programa independiente se le conoce como paso, **step**. Tras iniciar la ejecución de la cadena, Primero se ejecutará los pasos configurados como iniciales y a continuación los pasos dependientes del resultado de estos. Cada paso puede ejecutar uno de los siguientes elementos de *Scheduler*:

- Un Programa
- Un Evento

Un paso que apunte a un evento esperará hasta que se produzca el evento. Si se realiza el evento el paso se realizar completamente.

- Otra Cadena

Varios pasos de la misma cadena pueden hacer referencia al mismo programa o cadena.

En cierto sentido, una cadena se parece a un árbol de decisión, con muchos caminos posibles para seleccionar qué pasos has de ejecutar y cuando. Para ello se utiliza una lista de condiciones para decidir que acciones hay que realizar en cada paso. Un ejemplo de condición sería “Si el paso 2 falla o el paso 3 no funciona, espera una hora y inicia el paso 4.”.

Cuando se ejecuta una tarea asociada a una cadena, se puede monitorizar el estado de ejecución de cada uno de los pasos que forman la cadena.

Una situación típica en la que se puede hacer uso de cadenas son en las operaciones financieras, ya que precisan ejecutar diferentes programas para obtener un resultado final deseado.

Como están relacionados los programas, las tareas y las planificaciones

Para definir lo que se ejecuta y cuando, se asignan relaciones entre los programas, las tareas y las planificaciones. La ilustración1 muestra ejemplos de tales relaciones:

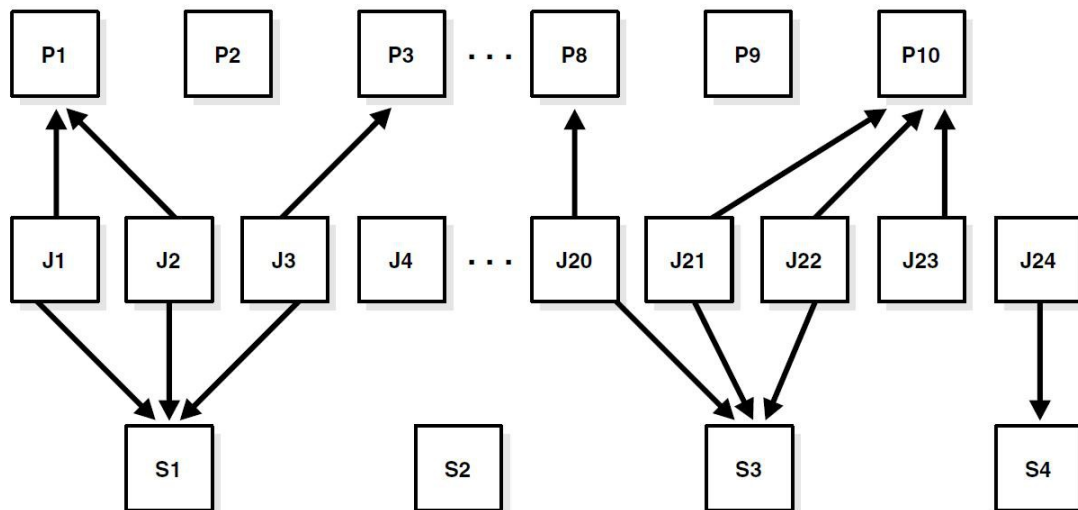


Ilustración 1: Relaciones entre los Programas, Tareas y Planificaciones

Para entender la *ilustración 1*, considere la siguiente situación: P1 es un programa para analizar una tabla utilizando el paquete DBMS_STATS. El programa tiene un parámetro de entrada, el nombre de la tabla a analizar. Las tareas, J1 y J2, utilizan este mismo programa, pero cada tarea asigna un nombre de tabla diferente. Además, la planificación S1 especifica un tiempo de ejecución de “Todos los días a las 2:00 am”. El resultado final es que las dos tablas especificadas en J1 y J2 son analizadas todos los días a las 2:00 am.

Nótese que la tarea J4 no tiene ningún tipo de relación, esto es debido a que toda la información



esta definida en la misma tarea. P2, P9 y S2 ilustran que se pueden dejar programas y planificaciones no asignadas. Se puede por ejemplo crear un programa que obtiene un inventario de fin de año y de forma temporal dejarlo de asignar a una tarea.

Conceptos Avanzados

Para los administradores de base de datos Scheduler ofrece un control para los contenidos más avanzados de la programación. Estos temas no son importantes para los que desarrollan aplicaciones.

Esta sección trata estos conceptos avanzados:

- **Job Classes** (Clases de Tarea)
- **Windows** (Ventanas de trabajo)
- **Windows Groups** (Grupo de ventanas de trabajo)

Clases de tareas

Las clases de tarea , **Job Classes**, proporcionan:

- Asignar un mismo conjunto de atributos a un grupo de tareas.
Cada clase especifica un conjunto de atributos, como el nivel del registro. Cuando se le asigna una clase a una tarea, esta hereda sus atributos.
- Establecer una serie de servicios a un grupo de tareas
A la clase se les puede asignar un servicio de la base de datos, heredando los recursos asignados al servicio.

- Establecer una serie de recursos a un grupo de tareas

Las clases sirven de vinculo entre los Desarrollares y los Administradores. En las clases se especifican una serie de recursos. Si un grupo de tareas esta asignada a la misma clase, estas consumirán los mismos recursos, pudiendo asignarle más recursos según necesidades.

Si una clase esta asociada a un servicio, no es preciso indicar los recursos, ya que hereda los recursos del servicio.

- Agrupar tareas para su priorización

Las clases permiten priorizar las tareas, para ello se le asigna en la tarea un valor de prioridad que va del 1 a 5. De este modo un grupo de tareas asignada a la clase , y que tiene la misma planificación, tiene prioridad el que tiene el valor más alto. Esto asegura que una tarea importante tiene prioridad sobre una que no lo es.

Si dos tareas tienen el mismo valor de prioridad, la tarea con fecha de inicio anterior tendrá más prioridad. Si no se le asigna ninguna prioridad a la tarea, por defecto tomará el valor 3.

- **NOTA:** Las prioridades solo tienen efecto sobre las tareas asignadas a una misma clase. No hay ninguna garantía que se inicie una tarea de prioridad alta en una clase A, antes que una tarea de prioridad baja de la clase B, aunque compartan la misma planificación. Dar prioridades a las tareas de diferentes clases dependerá del los recursos asignados a cada clase de tarea.

Ventanas de trabajo

Las ventanas de trabajo, **Windows**, se utilizan para cambiar la asignación de recursos entre las tareas durante varios periodo de tiempo, días, semanas y así sucesivamente. Una ventana de trabajo esta definido por un intervalo de tiempo con una fecha inicio y fecha fin bien definidos.

La ventanas trabajan con las clases para la asignación de recursos. Cada ventana de trabajo

especifica los recursos que se deben asignar al activarse. Las clases especifican un grupo de recursos o servicio de base de datos que tiene asignado su grupo de recursos. Por tanto, una tarea que se ejecuta dentro de una ventana dependerá de los recursos asignados a la clase y esta depende del plan de recursos de la ventana.

La *ilustración 2* muestra una jornada de trabajo que incluye dos ventanas de trabajo. En esta configuración las tareas asignadas al *Consumer Group 1* reciben más recursos por la mañana que por la tarde. Las tareas asignadas al *Consumer Group 2* les pasa todo lo contrario.

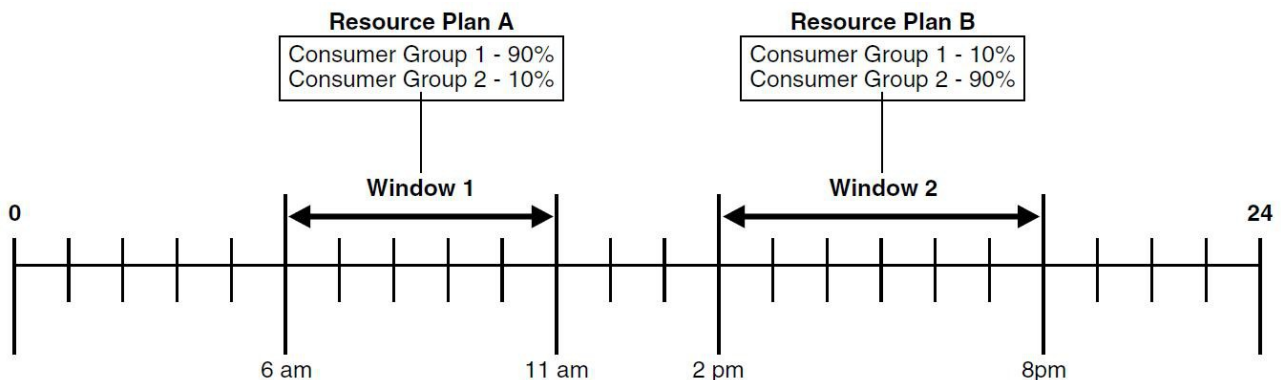


Ilustración 2: Ventana de trabajo para asignar recursos a las tareas

Se pueden dar prioridades a las ventanas. Si las ventanas se superponen, la ventana con mayor prioridad prevalecerá sobre las ventanas con prioridad mas baja. *Scheduler* abre y cierra las ventanas según los intervalos de tiempo que se establezcan.

Una tarea puede estar asociada a una ventana. De este modo, cuando *Scheduler* abre la ventana ejecutara la tarea. Si la tarea se crea mientras la ventana este abierta, no se ejecutará hasta que se vuelva a abrir la ventana.

Grupo de ventanas de trabajo

Los grupos de ventanas, **Windows Groups**, se utilizan para facilitar la planificación de las tareas. Si una tarea se debe ejecutar durante varios periodos de tiempo a lo largo del día, semana y así sucesivamente, se puede crear una ventana para cada periodo de tiempo y, a continuación, agregar las ventanas a un grupo de ventanas. Se puede establecer este grupo de ventanas a una tarea, de este modo la tarea se ejecutara durante todos los periodos de tiempo especificados en el grupo de ventanas.

Por ejemplo, si se tiene una ventana de trabajo llamada "Fines de Semana" y otra "Entre Semana", se puede crear un grupo de ventana que se llame "Semana". Se podría crear una tarea que utilice el grupo de ventana "Semana", de esta forma se ejecutara cuando se abre cada ventana asignada al grupo. Sabiendo siempre cuantos recursos se tiene asignado para la ejecución de la tarea.

Si una ventana de un grupo de ventanas ya esta abierto, y se crea una tarea que apunta a este grupo de ventanas, no se ejecutara hasta que la próxima ventana se abra.

Arquitectura

Esta sección trata de la arquitectura de *Scheduler* y se define:

- **The Job Table** (tabla de tareas)
- **The Job Coordinator** (coordinador de tareas)
- **How Jobs Execute** (Como se ejecutan las tareas)

- **Job Slaves** (Tareas esclavas)
- **Usando Scheduler en un entorno de aplicación real**

La *ilustración 3* muestra como las tareas están asociadas a la base de datos.

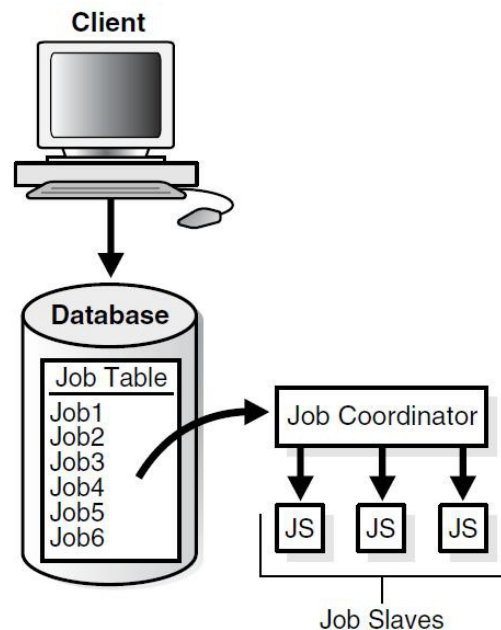


Ilustración 3: Componentes Scheduler

Tabla de tareas

La tabla de tareas, **Job Table**, almacena la información de todas las tareas, como el nombre del propietario, nombre de la tarea o el registro. Se encuentra disponible en todas las bases de datos y se puede consultar a partir de las vistas *_SCHEDULER_JOBS.

Las tareas son objetos de base de datos y se puede ir acumulando ocupando espacio. Para evitar esto, las tareas se eliminan tras finalizar. Esto se controla mediante el atributo de la tarea *auto_drop*.

Coordinador de tareas

El coordinador de tareas, **Job Coordinator**, es un proceso que se ejecuta en segundo plano (*cjqNNN*) que se inicia cuando se ejecuta una tarea o se activa una ventana de trabajo. Tras un periodo de inactividad largo en *Scheduler* este coordinador se detiene. El coordinador puede:

- Controlar y generar las tareas esclavas
- Consultar la tabla de tareas
- Recoge las tarea de las tabla y los almacena de forma regular en la memoria cache. Esto mejora el rendimiento y evita ir al disco.
- Recoge las tareas de la cache y los convierte en tareas esclava para su ejecución.
- Limpia las tareas esclavas cuando ya no son necesarias.



- Se pone en pausa cuando no hay tareas planificadas.
- Se activa cuando una tarea va a ser ejecutada o cuando se ha creado una nueva tarea.
- Al iniciar la base de datos, tras un cierre inesperado, recupera todas las tareas que estaban ejecutándose.

No es necesario indicar cuando el coordinador debe acceder a la tabla de tareas, el sistema selecciona este tiempo de forma automática.

Por cada instancia se utiliza un coordinador de tareas. Este es el funcionamiento que se utiliza en los entornos RAC.

Como se ejecutan las tareas

Cuando se selecciona una tarea para su ejecución, la tarea esclava:

- Reúne todos los datos para la ejecución. Por ejemplo, argumentos de programa o privilegios.
- Inicia una sesión con el propietario de la tarea, inicia la transacción, y a continuación inicia la ejecución de la tarea.
- Una vez finalizado la tarea, se guardan los cambios y finaliza la transacción.
- Cierra la sesión.

Tareas esclavas

Las tareas esclavas, **Job Slaves**, ejecutan las tareas que reciben. El coordinador activa un esclavo cuando hay que ejecutar una tarea. Y Este recoge los datos necesarios para ejecutar la tarea de la tabla.

Cuando se va a iniciar una tarea, los esclavos:

- Vuelven a planificar la tarea si es necesario.
- Actualiza el estado de la tabla para indicar si la tarea esta completado o si se tiene que volver a ejecutarse.
- Inserta en el registro de tareas
- Actualiza el numero de ejecuciones y si es necesario contar el numero de intentos y las veces que ha fallado.
- Limpiar
- Buscar una nueva tarea (de lo contrario se vuelve a deshabilitar)

Scheduler dinámicamente establece mas esclavos si es necesario.

Usando Scheduler en un entorno de aplicación real

En un entorno de aplicación real (RAC), *Scheduler* utiliza una tabla de tareas para cada base de datos y un coordinador de tareas por cada instancia. Los coordinadores se comunican entre si para mantener la información actualizada. *Scheduler* intenta equilibrar la carga de tareas de una clase en todas las instancias disponibles cuando la clase no tiene afinidad a ningún servicio, o a través de las instancias asignadas al servicio en particular, cuando la clase tiene definido el servicio.

La *ilustración 4* muestra una arquitectura típica de RAC, se puede ver como el coordinador de tareas de cada instancia intercambia información con los demás.

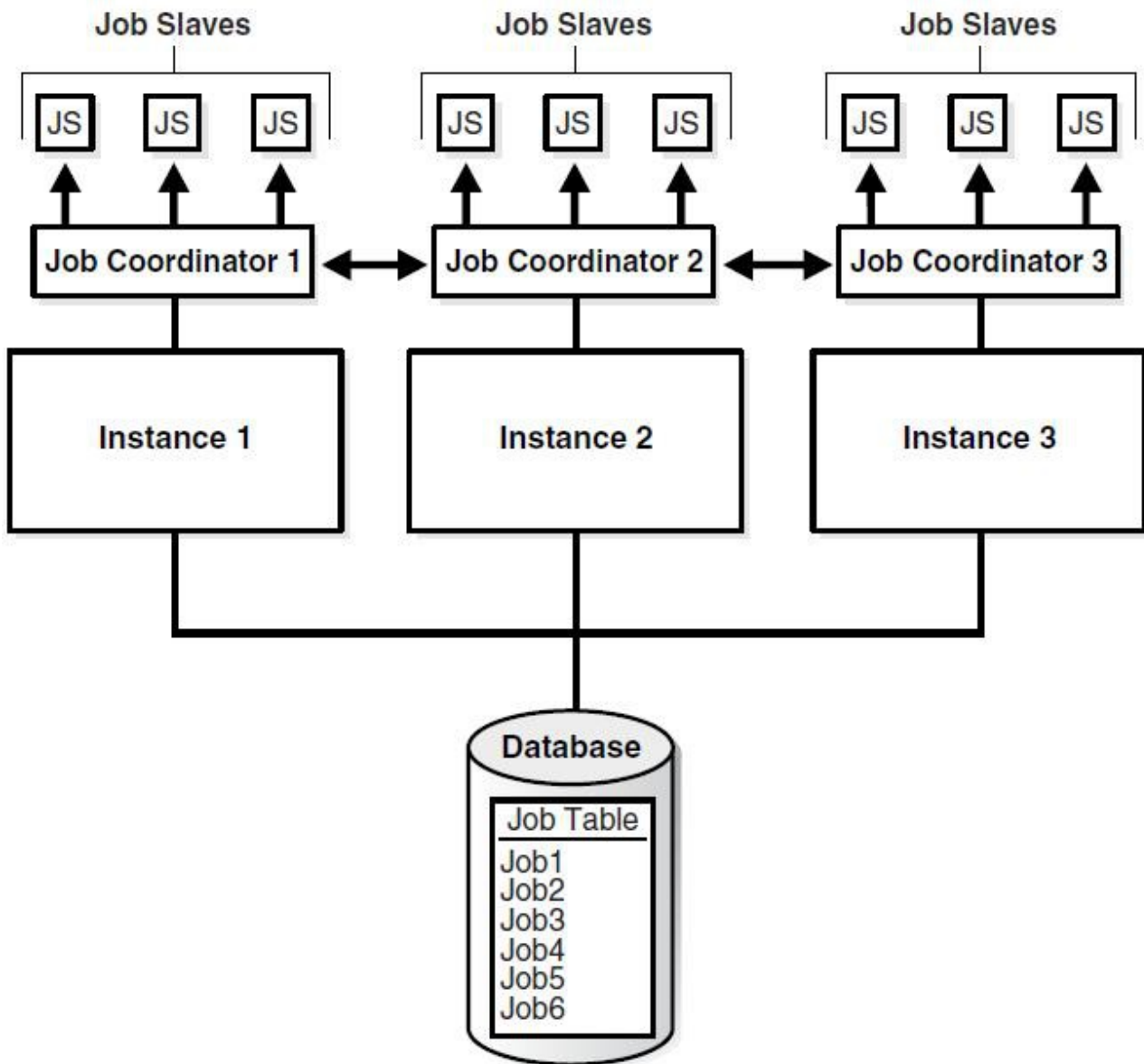


Ilustración 4: Arquitectura RAC en Scheduler

Como Funciona Scheduler cuando se tiene afinidad a un Servicio

En *Scheduler* se puede especificar al servicio de base de datos en el que se debe de ejecutar la tarea (afinidad a un servicio). Esto asegura una mayor disponibilidad de afinidad a la instancia, ya que garantiza que los demás nodos se pueden asignar de forma dinámica al servicio si una instancia deja de funcionar, ninguna de las tareas con afinidad a la instancia sera capaz de ejecutarse hasta que vuelva a funcionar. La ilustración 5 muestra un ejemplo de como se podrían utilizar los servicios y las instancias.

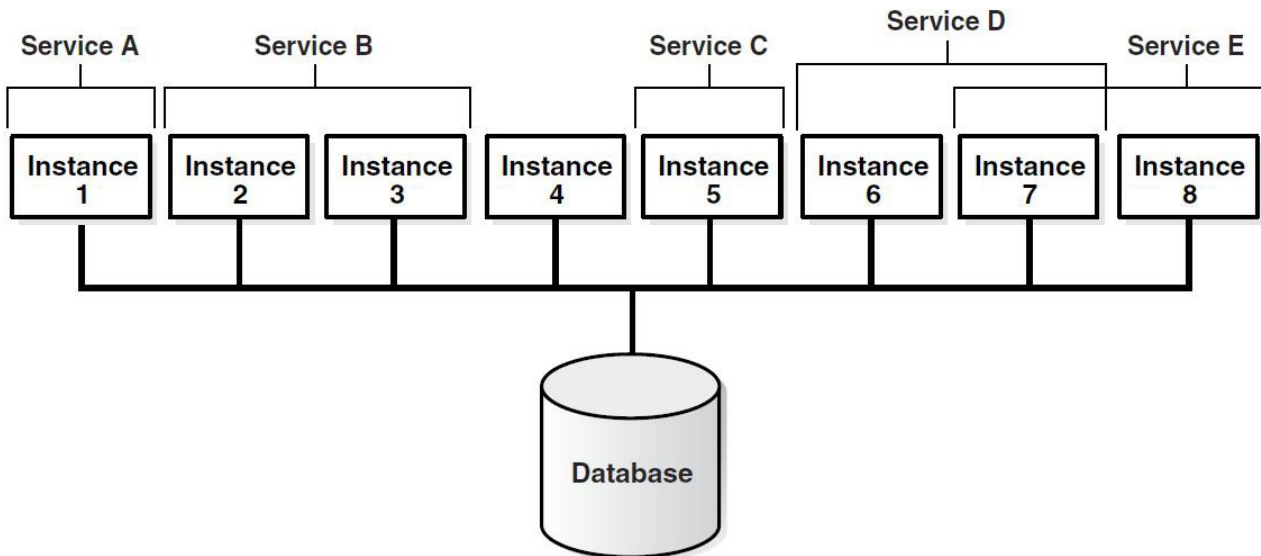


Ilustración 5: Scheduler, Afinidad a un Servicio

En la *ilustración 5*, si se cambian las propiedades de los servicios, *Scheduler* reconocerá automáticamente los cambios.

Cada clase puede especificar un servicio de base de datos. Si no se especifica ningún servicio, la clase pertenece a un servicio interno que garantiza la ejecución de la instancia.

3.3 Especificación Funcional

El objetivo de este punto es establecer la especificación funcional de una aplicación de gestión de tareas utilizando el paquete **dbms_scheduler**.

En la siguiente ilustración se muestra la estructura del sistema agrupado en módulos funcionales:

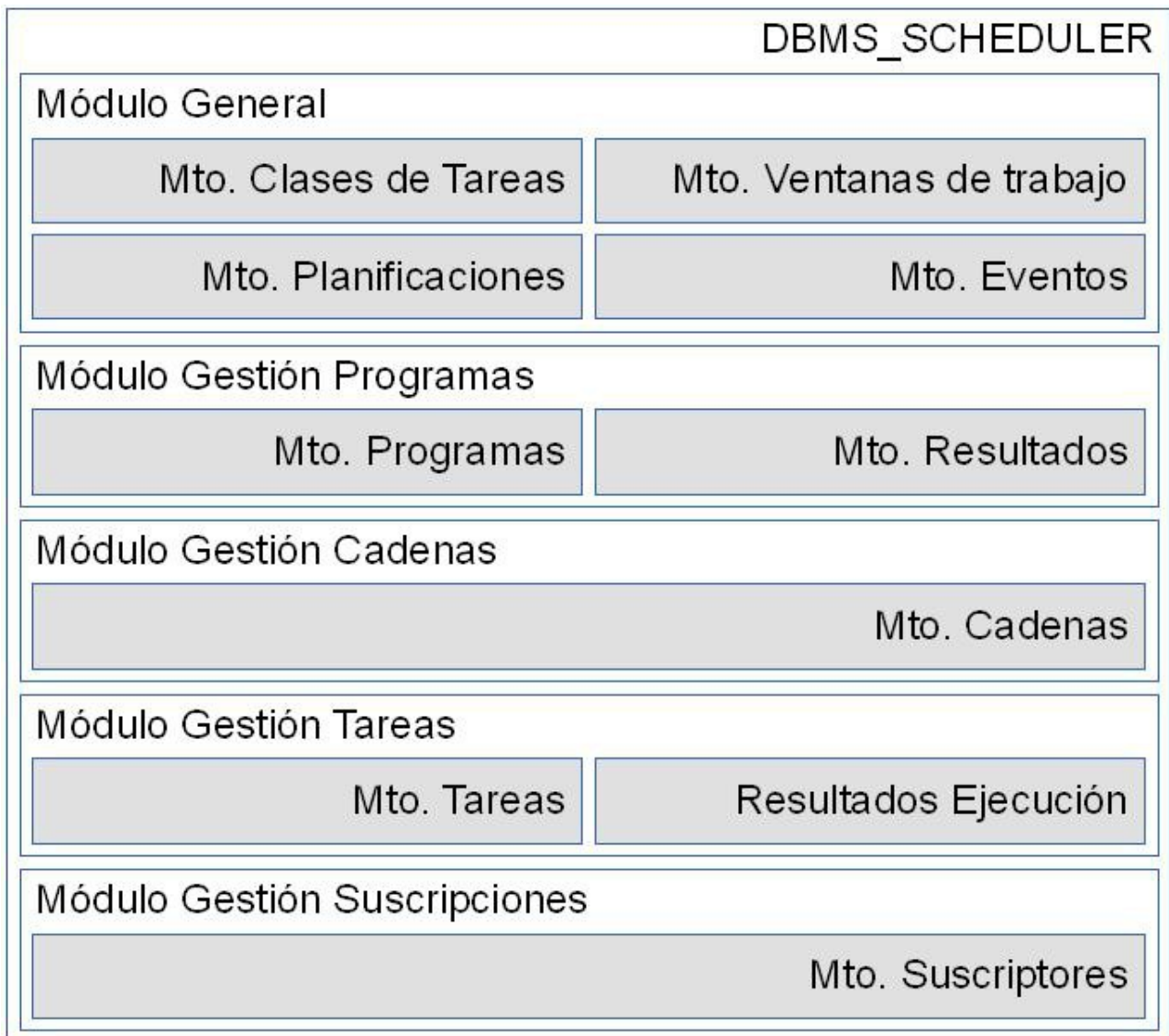


Ilustración 6: Estructura Aplicación DBMS_SCHEDULER

Módulo General

Este Modulo esta orientado para los Administradores de Scheduler, debido a que engloba los mantenimientos de todos los objetos más avanzados de Scheduler:

- Clases de tareas
- Ventanas de trabajo
- Planificaciones
- Eventos

Mantenimiento Clases de Tareas

Las Clases de Tareas (**Job Classes**) proporcionan:

- Asignar un mismo conjunto de atributos a un grupo de tareas.
- Establecer una serie de servicios a un grupo de tareas.
- Establecer una serie de recursos a un grupo de tareas.
- Agrupar tareas para su priorización.

En la siguiente tabla se describen los campos asociados a una Clase de Tarea indicando su condición de obligatoriedad y editabilidad.

Datos CLASE DE TAREA			
Campo	Descripción / Comentarios	Oblig	Edit
Nombre	Identificación de la Clase de Tarea	S	N
Descripción	Descripción de la Clase de Solicitud	S	S
Nivel de log	Determina la información del log que se registra. Permite dos opciones: <ul style="list-style-type: none">• LOG EJECUCIÓN: Se registra información asociada solo a la ejecución.• LOG COMPLETO: Se registra toda la información de todas las Tareas asociadas a esta Clase de Tarea.	S	S
Recursos	Determina el Grupo de Consumo de Recursos que se asociara a la Clase de Tarea. Este campo es Excluyente con el campo Servicio.	N	S
Servicio	Determina el Servicio de BD que se asociara la Clase de Tarea. Este campo es Excluyente con el campo Recursos.	N	S
Historial	Determina el número de días que permanece los registros de log en el sistema antes de ser eliminados.	S	S

Tabla 1: Datos Clase de Tarea

Descripción Funcional

Este mantenimiento debe permitir:

- **Crear** Clases de Tarea.

Para crear una nueva Clase de Tarea se deben cumplir las siguientes validaciones:



- Campos obligatorios informados.
- No debe existir ninguna Clase de Tarea con el mismo nombre.
- Debe informarse el Servicio o el Grupo de Consumo
- **Modificar** Clases de Tarea.
Para modificar una Clase de Tarea se deben cumplir las siguientes validaciones:
 - Campos obligatorios informados.
 - No debe existir ninguna Clase de Tarea con el mismo nombre
 - Debe informarse el Servicio o el Grupo de Consumo
- **Eliminar** Clases de Tarea.
Para eliminar una Clase de Tarea se deben cumplir las siguientes validaciones:
 - No debe existir ninguna Tarea que haga referencia a la Clase de Tarea a eliminar.

Mantenimiento Ventanas de trabajo

Las Ventanas de Trabajo permiten la activación de recursos en diferentes periodos temporales conforme a las necesidades de los administradores del sistema. Las ventanas definen los recursos disponibles para los procesos que se ejecutan mientras estas estén abiertas. En la siguiente tabla se describen los campos asociados a las Ventanas indicando su condición de obligatoriedad y editabilidad.

Datos VENTANA DE TRABAJO			
Campo	Descripción	Oblig	Edit
Nombre	Identificador de la Ventana de ejecución	S	N
Plan Recurso	Plan de recurso que se activa automáticamente cuando se inicia la ventana de trabajo	S	S
Fecha inicio	Indica la primera fecha en la que la ventana se planifica para abrirse	N	S
Duración	Determina el tiempo que permanece abierto la ventana de trabajo	S	S
Intervalo Ejecución	Determina la periodicidad en que la ventana se volverá a abrir.	S	S
Fecha fin	Indica la fecha tras la cual la ventana no se abrirá más	N	S
Prioridad	Prioridad cuando dos ventanas se superponen.	S	S
Próximo inicio	Indica la próxima ocasión en que la ventana se abrirá	N	N
Próximo Fin	Indica la próxima ocasión en que la ventana se cerrará	N	N
Activa	Indica si la ventana esta activa o desactiva	N	S

Tabla 2: Datos Ventana de Trabajo

Descripción Funcional

Este mantenimiento debe permitir:

- **Crear** Ventanas de Trabajo.

Para crear una nueva Ventana de Trabajo se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- No debe existir ninguna Ventana con el mismo nombre.
- Debe informarse un Plan de Recurso valido.

- **Modificar** Ventanas de Trabajo.

Para modificar una Ventanas de Trabajo se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- No debe existir ninguna Ventana con el mismo nombre
- Debe informarse un Plan de Recurso valido.

- **Eliminar** Ventanas de Trabajo.

Para eliminar una Ventanas de Trabajo se deben cumplir las siguientes validaciones:

- No debe existir ninguna Tarea que haga referencia a la Ventana a eliminar.

Mantenimiento Planificaciones

Las planificaciones son las codificaciones de los hitos temporales y periódicos. Estas codificaciones se podrán asignar a una Tarea o a una Ventana de Ejecución.

En la siguiente tabla se describen los campos asociados a las Planificaciones indicando su condición de obligatoriedad y editabilidad.

Datos PLANIFICACIONES			
Campo	Descripción	Oblig	Edit
Nombre	Identificador de la Planificación	S	N
Descripción	Descripción de la planificación	N	S
Fecha inicio	Indica la fecha a partir de la cual la planificación es valida.	N	S
Intervalo Ejecución	Determina la periodicidad con la que debe de ejecutarse la tarea o abrirse la Ventana de Trabajo. Para ello se deberá determinar las siguientes características: <ul style="list-style-type: none">• Frecuencia• Intervalo Repetición• Clausulas adicionales	S	S
Fecha fin	Indica la fecha a partir de la cual la planificación no es valida. Si una tarea tiene asociada la planificación, no se volverá a ejecutar. Si una Ventana tiene asociada la planificación, no se volverá a abrir.	N	S

Tabla 3: Datos Planificaciones

La información para el campo de **Intervalo Ejecución** se podrá introducir de forma manual utilizando la sintaxis de Scheduler para la definición de planificaciones o bien utilizar un Formulario de *Detalle de Planificación* que guiará al usuario en la introducción de datos.

Descripción Funcional

Este mantenimiento debe permitir:

- **Crear** Planificaciones.

Para crear una nueva Planificación se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- No debe existir ninguna Planificación con el mismo nombre.

- **Modificar** Planificaciones.

Para modificar una Planificación se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- No debe existir ninguna Planificación con el mismo nombre

- **Eliminar** Planificaciones.

Para eliminar una Planificación se deben cumplir las siguientes validaciones:

- No debe existir ninguna Ventana o Tarea que haga referencia a la Planificación.

- **Evaluar** Planificaciones.



Esta funcionalidad permite consultar las próximas fechas a las que hace referencia una planificación para validar la correcta definición de la misma.

Para evaluar una Planificación se deben cumplir las siguientes validaciones:

- Indicar el número de próximas ejecuciones que se desea mostrar.



Mantenimiento Eventos

Un evento es un mensaje informativo enviado al Scheduler, por una aplicación externa, para comunicarle cualquier acontecimiento producido en el sistema. La recepción de este mensaje puede desencadenar la ejecución de un proceso específico encargado de gestionar dicho mensaje.

En la siguiente tabla se describen los campos asociados a los Eventos indicando su condición de obligatoriedad y editabilidad.

Datos EVENTOS EXTERNOS			
Campo	Descripción	Oblig	Edit
Nombre	Identificador del Evento	S	N
Descripción	Descripción del Evento	N	S
Fecha inicio	Fecha a partir de la cual el Evento es válido. Eventos anteriores serán ignorados	N	S
Cola	Nombre de la cola donde las aplicaciones externas incluirán sus mensajes para que sean gestionados por Scheduler.	S	S
Condición	Expresión lógica que determina cuando se produce el Evento.	S	S
Fecha fin	Fecha partir de la cual el Evento expira.	N	S

Tabla 4: Datos Eventos Externos

Descripción Funcional

Este mantenimiento debe permitir:

- **Crear** Eventos.

Para crear un nuevo Evento se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- No debe existir ningún Evento con el mismo nombre.

- **Modificar** Eventos.

Para modificar un Evento se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- No debe existir ningún Evento con el mismo nombre
- No debe estar asociado a ninguna Cadena o Tarea.

- **Eliminar** Eventos.

Para eliminar un Evento se deben cumplir las siguientes validaciones:

- No debe estar asociado a ninguna Cadena o Tarea.



Módulo Gestión Programas

Este modulo se ocupa de la gestión de los programas (**Program Scheduler**) y de la gestión de resultados del programa.

Este nuevo concepto de **Resultado** surge por la necesidad de crear programas que ejecuten informes o que los creen directamente, para que puedan ser enviados posteriormente a un conjunto de usuarios.

Mantenimiento Programas

Un programa define la estructura de un componente ejecutable encargado de realizar una acción cuando se planifique su ejecución a través de una *Tarea*.

Este mantenimiento debe permitir la navegación entre los distintos programas de una aplicación. Para cada programa deben estar disponibles las siguientes opciones.

- Datos Generales. Atributos específicos del programa.
- Parámetros del Programa. Parámetros o Argumentos asociados a un programa.
- Resultados del Programa. Ficheros que se crear tras la ejecución del programa.

En los **Datos Generales** se definen aquellos atributos que identificarán al programa como el Nombre del Programa, Descripción y el responsable del programa entre otros ([ver Tabla 5](#)).

Los tipos de programa que se pueden catalogar son:

- Procedimientos almacenados en la Base de datos
- Procedimientos externos (Shells Script)
- PL/SQL

Los estados contemplados en el ciclo de vida de un programa son:

- Activo: Permite la ejecución del programa a través de la planificación de una *Tarea*.
- Desactivo: No permitirá la ejecución de las *Tareas* que referencia al Programa.

Un programa podrá desactivarse manualmente o bien de forma automática si la estructura del programa es inconsistente.

En los **Parámetros del Programa** se especifican los argumentos de entrada del programa (si los tiene), en caso contrario la estructura del programa será inconsistente y el estado del programa será **Desactivo** ([ver Tabla 6](#)).

Solo se admiten parámetros de entrada pertenecientes a alguno de los siguientes tipos:

- Carácter
- Numérico
- Fecha

En los **Resultados del Programa** se indican los ficheros resultado de la ejecución de un programa, se podrá generar uno o varios ficheros ([ver Tabla 7](#)).

En las siguientes tablas se describen los campos asociados a los Programas, Parámetros de Programa y Resultados de Programa indicando su condición de obligatoriedad y editabilidad.



Datos PROGRAMAS			
Campo	Descripción	Oblig	Edit
Nombre	Identificador del Programa	S	N
Descripción	Descripción del Programa	S	S
Tipo	Tipo de programa: <ul style="list-style-type: none">• Procedimiento almacenado• Ejecutable externo• PL/SQL	S	N
Acción	Acción que realiza el programa. Si se trata de un Ejecutable externo (Shell Script o Aplicación Java) hay que indicar el directorio.	S	N
Número de Argumentos	Número de parámetros o argumentos (Si se trata de un procedimiento almacenado, se recuperará automáticamente.)	N	N
Estado	Activo o Desactivo	S	S
Aplicación	Propietario al que pertenece el programa.	S	N
Responsable	Nombre del responsable del programa.	S	S
Tiempo estimado para la ejecución	Tiempo estimado de duración del programa expresado en HH:MI:SS. Se trata de un dato informativo, utilizado para estadísticas de rendimiento.	N	S

Tabla 5: Datos Programas

Datos PARÁMETROS DE PROGRAMAS			
Campo	Descripción	Oblig	Edit
Programa	Identificador del Programa	S	N
Nombre	Nombre del parámetro	S	S
Posición	Posición del parámetro en el programa	S	S
Tipo	Tipo de parámetro (Numérico, Carácter o Fecha)	S	S
Valor por defecto	Valor que se asociará por defecto al parámetro	N	S

Tabla 6: Datos Parámetros Programa

Datos RESULTADOS DE PROGRAMAS			
Campo	Descripción	Oblig	Edit
Programa	Identificador del Programa	S	N
Identificador	Identificador del fichero	S	N
Nombre	Nombre del fichero	N	S
Función	Función a ejecutar para obtener el nombre del fichero	N	S
Tipo	Tipo de Fichero (Log Ejecución o Informe)	S	S
Directorio	Directorio físico del fichero	S	S
Borrado	Indicador de borrado del fichero tras su envío	S	S

Tabla 7: Datos Resultados de Programa



Descripción Funcional

Este mantenimiento debe permitir:

- **Crear** Programas

Para crear un nuevo Programa se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- No debe existir ningún Programa con el mismo nombre.
- Debe existir la aplicación. Validación sobre la tabla de aplicaciones.
- Debe existir el Responsable. Validación sobre la tabla de usuarios.
- Número de Argumentos, depende del tipo de programa:
 - PL/SQL: Numero Argumentos es cero
 - Procedimiento Almacenado: Obtener Número Argumentos de las tablas del Sistema.
 - Ejecutable Externo: Introducir Número Argumentos manualmente.
- **Definición de parámetros** Programa:

Solo se permite definir los Parámetros para programas de tipo *Procedimiento Almacenado*, si tiene parámetros de entrada, y *Ejecutables Externos*, si se ha indicado el Número de Argumentos.

En el caso de los *Procedimientos Almacenados* los parámetros se deben crear de forma automática, no permitiendo incluir nuevos parámetros ni modificar los ya existentes. El único campo que se permite modificar es el *Valor por Defecto*.

En el caso de *Ejecutables Externos* se permiten crear tantos parámetros como se ha indicado en el campo *Número de Argumentos* del Programa.

Para crear un parámetros deben cumplirse las siguientes validaciones:

- Campos obligatorios informados.
- No debe existir otro Parámetro con la misma posición.
- **Definición de resultados** Programa:

Para crear Ficheros de Resultado deben cumplirse las siguientes validaciones:

 - Campos obligatorios informados.

- **Modificar** Programas.

Para modificar un Programa se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- Debe existir la aplicación. Validación sobre la tabla de aplicaciones.
- Debe existir el Responsable. Validación sobre la tabla de usuarios.
- No debe estar asociado en ninguna Cadena o Tarea.
- **Modificación parámetros** Programa:

En los Programas de tipo *Procedimiento Almacenado* el único campo modificable es *Valor por Defecto*.

En los Programas de tipo *Ejecutable Externo* se pueden modificar todos los campos.



Para modificar un parámetro se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- No debe existir otro parámetro con la misma posición.
- Para modificar cualquier campos que no sea el *Valor por Defecto*, el Programa al que pertenece el parámetro no puede estar asociada a ninguna Cadena o Tarea.

○ **Modificación resultados** Programa:

Para modificar un resultado se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.

• **Eliminar** Programas.

La eliminación de un programa conlleva la eliminación de los Parámetros y Resultados asociados.

Para eliminar un Programa se deben cumplir las siguientes validaciones:

- No debe estar asociado a ninguna Cadena o Tarea.

○ **Eliminar Parámetros** Programa:

En los programas de tipo *Procedimiento Almacenado* no se permite eliminar ningún parámetro.

En los programas de tipo *Ejecutable Externo* si se permite la eliminación de parámetros. Para ello se deben cumplir las siguientes validaciones:

- El programa al que pertenece el parámetro no puede estar asociada a ninguna Cadena o Tarea.

○ **Eliminar Resultados** Programa:

Para eliminar un fichero se deben cumplir las siguientes validaciones:

- No deben existir suscripciones al Fichero.

• **Activar y Desactivar Programas**

Para **Activar** un programa, y así permitir que sea reverenciado en una Cadena o Tarea el programa debe estar correctamente creado.

Para **Desactivar** un programa, este no debe estar asociado a ninguna Cadena o Tarea.

• **Creación Cadena Simple.**

Una vez creado el programa se permitirá crear una Cadena con un único paso que referencia al programa seleccionado.

• **Ejecución Inmediata.**

La ejecución inmediata consiste en la ejecución del programa seleccionado en el momento en el que se realiza la petición.

• **Consultas y Listados.**

Muestra una lista con las Tareas y Cadenas asociadas al programa seleccionado. Para las Cadenas se mostrará una nueva lista con las Tareas asociadas.

Módulo Gestión Cadenas

Este Modulo se ocupa del Mantenimiento y de la Gestión de las Cadenas (**Chains Scheduler**). Permite la navegación entre las distintas cadenas de una aplicación.

Mantenimiento Cadenas

Una cadena es una agrupación de objetos Scheduler (Programas, Eventos o otros procesos) con dependencias entre ellas, y una lógica de aplicación entre las mismas.

Este mantenimiento debe permitir la navegación entre las distintas cadenas de una aplicación. Para cada cadena se debe definir los siguientes atributos:

- Datos Generales: Atributos específicos de la cadena.
- Pasos : Pasos asociados a la cadena.
- Dependencias: Condiciones de ejecución de los Pasos.

En los **Datos Generales** se definen aquellos atributos que identificarán a la Cadena como el nombre, aplicación al que pertenece y el responsable de la cadena (Ver tabla 8).

Los **Pasos** forman parte de la definición de una Cadena. Un paso puede hacer referencia a:

- Un Programa o Cadena previamente definido en el sistema. Cuando se cumplan las condiciones específicas se ejecutará el programa o proceso asociado.
- Un Evento. En este caso, el paso se quedará a la espera de que se produzca el evento, para su ejecución.

En el esquema siguiente se muestra los **estados** en los que se puede encontrar un paso:

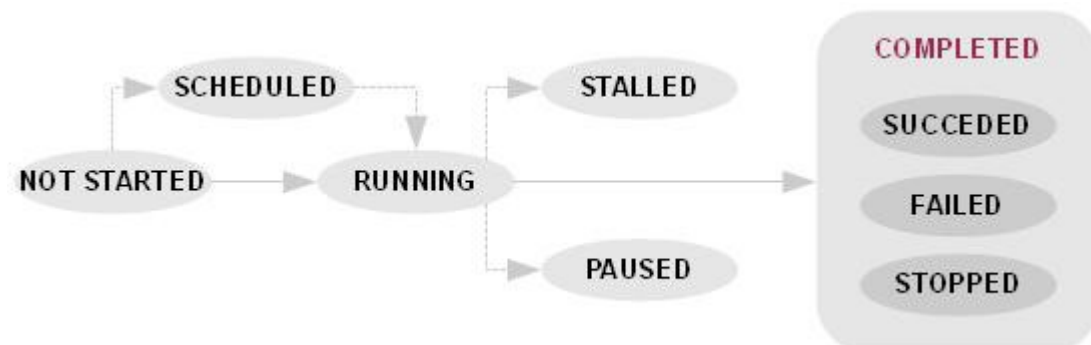


Ilustración 7: Estados de los Pasos de una Cadena

El significado de cada uno de los estados se describe en la siguiente tabla:



Estados de un Paso	
Estado	Descripción
NOT STARTED	Sin iniciar. La cadena al que pertenece el paso se esta ejecutando, pero el paso no ha comenzado su ejecución.
SCHEDULED	Planificado. El paso definido con la clausula AFTER provoca que este comience pero el tiempo designado por dicha clausula aun no ha expirado.
RUNNING	En ejecución. Se esta ejecutando el paso. Para un paso asociado a un evento, el paso a comenzado y está esperando a que se produzca el evento.
PAUSED	Pausado. El paso ha sido interrumpido.
SUCCEDED	Correcto. El paso se ha completado correctamente.
FAILED	Incorrecto. El paso se ha completado pero con errores.
STOPPED	Parado. El paso ha sido parado mediante la ejecución de Parara Cadena.
STALLED	Atascado. La acción asociada al paso esta atascado. Precisa de intervención manual.

Tabla 8: Estados de los Pasos de una Cadena

Una vez definidos los pasos de una cadena se deben definir las **dependencias y condiciones de ejecución** de los mismos.

En los esquemas siguientes se representan dos ejemplos de las posibilidades que puede ofrecer scheduler en cuento a la definición de las dependencias entre los pasos de una cadena:

Para que comience la ejecución de una cadena, debe existir al menos un paso inicial. Se entiende como paso inicial, aquel que no depende de ningún otro paso de la cadena y que su condición de ejecución se evalúa siempre a verdadero.

Del mismo modo, para que una cadena finalice su ejecución es necesario que contenga algún paso final. Se entiende como paso final, aquel para el que exista una dependencia que indique que la cadena finaliza cuando lo hace el paso.

Con el objetivo de asegurar la correcta definición de las cadenas, junto con sus pasos y dependencias se establece la siguiente casuística para el mantenimiento de dependencias:

- Para cada cadena se define al menos un paso inicial.
- Para cada paso no inicial se establece una única condición que puede ser tan compleja como se precise utilizando las funciones lógicas AND y OR.
- Para cada cadena se establece al menos una condición que determina el conjunto de validaciones que deben cumplirse para la finalización de dicha cadena.

En las siguientes tablas se describen los campos asociados a la Cadena y Pasos indicando su condición de obligatoriedad y editabilidad.



Datos CADENA			
Campo	Descripción	Oblig	Edit
Nombre	Identificador de la Cadena	S	N
Descripción	Descripción de la Cadena	S	S
Estado	Estado de la Cadena: Activo o Desactivo.	S	S
Aplicación	Propietario al que pertenece la Cadena.	S	S
Responsable	Nombre del responsable de la cadena.	S	S
Condición Finalización	Conjunto de reglas que deben cumplirse para considerar la cadena como finalizada tras su ejecución.	N	S

Tabla 9: Datos Cadena

Datos PASOS			
Campo	Descripción	Oblig	Edit
Cadena	Identificador de la Cadena	S	N
Nombre	Identificador del Paso	S	N
Acción	Programa o Cadena (ya existente) asociado al paso Este campo es Excluyente con el campo Evento	N	S
Evento	Evento (ya existente) asociado al paso Este campo es Excluyente con el campo Acción	N	S
Condición	Condición de ejecución del paso	S	S

Tabla 10: Datos Pasos de una Cadena

Descripción Funcional

Este mantenimiento debe permitir:

- **Crear** Cadena.

Para crear una nueva Cadena se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- No debe existir ninguna Cadena con el mismo nombre.
- Debe existir la aplicación. Validación sobre la tabla de aplicaciones.
- Debe existir el Responsable. Validación sobre la tabla de usuarios.
- **Definición Pasos:**
 - Para Crear nuevos Pasos se deben cumplir las siguientes validaciones:
 - Campos obligatorios informados.
 - No debe existir otro Paso asociado a la Cadena con el mismo Nombre.
 - Validar existencia de los Programas, Cadenas y Eventos asociados al paso.
 - La Cadena asociada al paso no debe estar asociada a ninguna Cadena o Tarea.



- **Definición Condiciones Dependencias:**
 - Para crear condiciones de dependencia se deben cumplir las siguientes validaciones:
 - Deben estar creados todos los pasos.
 - Al menos un Paso se definirá como Inicial.
- **Definición de Condiciones Finalización:**
 - Para crear condiciones de finalización se deben cumplir las siguientes validaciones:
 - Deben estar creados todos los pasos.
 - Al menos un paso se definirá como inicial.
- **Modificar Cadena.**

Para modificar una Cadena se deben cumplir las siguientes validaciones:

 - Campos obligatorios informados.
 - Debe existir la aplicación. Validación sobre la tabla de aplicaciones.
 - Debe existir el Responsable. Validación sobre la tabla de usuarios.
 - No debe estar asociado a ninguna Cadena o Tarea.
 - **Modificar Pasos:**
 - Para modificar un Paso se deben cumplir las siguientes validaciones:
 - Campos obligatorios informados.
 - No debe existir otro Paso asociado a la Cadena con el mismo Nombre.
 - Validar existencia de los Programas, Cadenas y Eventos asociados al paso.
 - La Cadena asociada al paso no debe estar asociada a ninguna Cadena o Tarea.
 - **Modificar Condiciones Dependencias:**
 - Para modificar condiciones de dependencia se deben cumplir las siguientes validaciones:
 - Deben estar creados todos los pasos.
 - Al menos un Paso se definirá como Inicial.
 - **Modificar Condiciones Finalización:**
 - Para modificar condiciones de finalización se deben cumplir las siguientes validaciones:
 - Deben estar creados todos los pasos.
 - Al menos un Paso se definirá como Inicial.
- **Eliminar Cadena.**

Para Eliminar una Cadena se deben cumplir las siguientes validaciones:

 - No debe estar asociado a ninguna Cadena o Tarea.
 - **Eliminar Pasos:**
 - Para eliminar un Paso se deben cumplir las siguientes validaciones:
 - La Cadena asociada al paso no debe estar asociada a ninguna Cadena o Tarea.



- El Paso no puede estar referenciado en ninguna Condición de Dependencia y de Finalización.
- **Activar** Cadena.
Para Activar una cadena y así permitir que sea referenciado en una Tarea, deben cumplirse las siguientes validaciones:
 - Debe existir al menos un paso definido en la Cadena.
 - Debe existir al menos una condición de ejecución del Paso Inicial.
 - Debe existir una condición de finalización de la Cadena.
- **Desactivar** Cadena.
Para Desactivar una Cadena, esta no debe estar asociado a ninguna Cadena o Tarea.
- **Copiar** Cadena.
Esta funcionalidad permite crear una nueva Cadena copiando todos los datos de la Cadena seleccionada y de los Pasos y Condiciones de ejecución.
- **Ejecución Inmediata**
La ejecución inmediata consiste en la ejecución de la Cadena en el momento en que se realice la petición. Para ello la Cadena debe estar Activa.
- **Consultas y Listados**
Muestra una lista con las Tareas y Cadenas asociadas a la Cadena seleccionada. Para las Cadenas se mostrará una nueva lista con las Tareas asociadas.



Módulo Gestión Tareas

Este Modulo se encarga del mantenimiento y de la gestión de Tareas (**Jobs Scheduler**). Permite la navegación entre las distintas tareas de una aplicación, asumiendo que la aplicación de una tarea es la aplicación del programa o proceso asociado.

Este modulo esta estructurado en los siguientes apartados:

- Mantenimiento de tareas

Esta opción permite dar de Alta, Modificar y Eliminar Tareas y los datos asociados a la misma.

- Gestión de tareas

Esta opción permite administrar las tareas existentes en el sistema.

Mantenimiento Tareas

Una tarea es una combinación de un Programa o una Cadena y una Planificación para la ejecución de los mismos.

Las tareas son los objetos a través de los cuales se realiza la ejecución de las acciones requeridos por cada aplicación.

Al finalizar la ejecución de una tarea se generará un evento que lo utiliza el **Gestor de Suscripciones** para activar el envío de correos.

La **planificación de una tarea** determina el momento en que se debe ejecutar. Existen distintos tipos de planificaciones:

- Ejecución Inmediata: Planifica la tarea para que se ejecute lo antes posible.
- Ejecución asociada a un Evento: Planifica la tarea para que su condición de ejecución se condicione a la ocurrencia de un evento externo. Dicho evento debe existir en el catalogo de eventos de Scheduler.
- Ejecución asociada a una Planificación: Planifica la tarea para que su condición de ejecución se condicione a una Planificación. Dicha planificación puede existir o no el catalogo de planificaciones de Scheduler.

En el esquema se muestra los **estados** en los que se puede encontrar una tarea. Aquellas tareas planificadas como Inmediatas, al finalizar la ejecución su estado corresponderá con el estado de la ejecución. En caso contrario, cuando estén previstas más de una ejecución, al finalizar la ejecución de la tarea su estado volverá al estado SCHEDULED si existen más ejecuciones previstas o COMPLETED si se trata de la última ejecución. Se puede consultar los estados de finalización de cada una de las ejecuciones de una tarea.

El significado de cada uno de los estados se describe en la tabla, junto con la siguiente información:

- Resultado Ejecución: Indica si el estado de la tarea se corresponde también con el estado final de la ejecución.
- Estado de los Pasos: Indica los estados en que puede encontrarse los pasos de una cadena en el estado actual de la tarea.

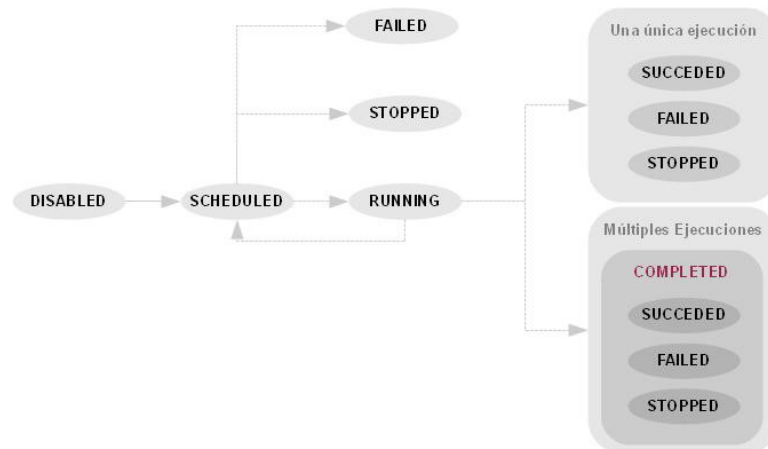


Ilustración 8: Estados de una Cadena

Estados de una Cadena			
Estado	Descripción	Resultado Ejecución	Estado Pasos
DISABLED	Deshabilitada. La tarea esta deshabilitada		
SCHEDULED	Planificada. La tarea esta planificada para su ejecución		
RUNNING	En ejecución. La tarea se esta ejecutando.		Cualquier estado excepto STALLED
COMPLETED	Finalizado. La tarea a finalizado y no esta planificada otra ejecución.		Todos los pasos están Finalizados (SUCCEEDED, FAILED, STOPPED)
STOPPED	Parada. La tarea se ha parado mientras se estaba ejecutando.	SI	Al menos un paso en estado STOPPED
BROKEN	Interrumpida. La tarea ha sido deshabilitada.		Al menos un paso en estado FAILED
FAILED	Incorrecta. La tarea ha finalizado con error.	SI	Al menos un paso en estado FAILED
SUCCEEDED	Correcta. La tarea se ha planificado para una sola ejecución y ha finalizado correctamente.	SI	Todos los pasos en estado SUCCEEDED
CHAIN_STALLED	Proceso Atascado. La cadena asociada se encuentra en estado STALLED.		Al menos un paso en estado STALLED

Tabla 11: Estados de las Tareas



La *Gestión de Tareas* permite realizar siguientes acciones sobre las Tareas ya creadas:

- Activar o Desactivar una tarea:

Modificar el estado de una Tarea para permitir su planificación y ejecución.

- Solicitar ejecución inmediata de una tarea:

Realizar una petición para que una Tarea se planifique para su ejecución inmediata.

- Monitorizar ejecución:

Revisar la situación en que se encuentra la ejecución de una tarea y en caso necesario realizar alguna intervención manual sobre la misma:

- Completar Paso.

Permite modificar el estado de un paso para alterar la ejecución de una Tarea. Es útil para resolver situaciones en la que los errores en la ejecución de uno de los pasos de una cadena impida la ejecución de los pasos sucesores, y se requiera, tras una intervención manual para solucionar los errores surgidos, continuar con la ejecución de dichos pasos.

- Parar Tarea.

Permite modificar el estado de una Tarea a STOPPED para finalizar su ejecución.

- Consultar dependencias.

Esta opción permite consultar la definición de una Cadena y de las dependencias de sus pasos.

- Revisar Resultados.

Permite consultar los estados de finalización de cada una de las ejecuciones de una tarea.

En las siguientes tablas se describen los campos asociados a una Tarea indicando su condición de obligatoriedad y editabilidad.



Datos TAREA			
Campo	Descripción	Oblig	Edit
Nombre	Identificador de la Tarea	S	N
Descripción	Descripción de la Tarea	S	S
Programa Cadena	Nombre del Programa o Cadena	S	S
Estado	Activo o Desactivo. El estado inicial es Desactivo. Una tarea podrá cambiar de estado según petición de usuario. Cambiará a estado Desactivo automáticamente si se detecta una inconsistencia en los componentes de la Tarea.	S	S
Clase Solicitud	Grupo de consumo de recursos.	S	S
Prioridad	Prioridad de la tarea respecto a las otras tareas pertenecientes a la misma clase de solicitud.	N	S
Planificación:			
Evento	Indica que la tarea esta asociada a un Evento Permite definir: <ul style="list-style-type: none"> Nombre Evento (Existente) Fecha Inicio: Fecha en que la tarea empezará a ejecutarse. Fecha Fin: Fecha en que la tarea dejará de ejecutarse. 	N	S
Predefinida	Indica que la tarea esta asociada una Planificación predefinida en el Mantenimiento de Eventos	N	S
Especifica	Indica que la tarea esta asociada a una Planificación que no esta predefinida. Teniendo que determinar las siguientes características: <ul style="list-style-type: none"> Frecuencia Fecha Inicio Fecha Fin Clausulas adicionales 	N	S

Tabla 12: Datos Tarea

Descripción Funcional

Este mantenimiento debe permitir:

- **Crear** Tareas.

Para crear una nueva Tarea se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- No debe existir ninguna Tarea con el mismo nombre.
- Los elementos asociados a la Tarea como: Programa, Cadena, Clases de Tarea y la Planificación Predefinida deben existir previamente.
- **Definición Parámetros** Tarea

Las Tareas asociadas a Programas se podrá definir el *Valor por Defecto* de los parámetros a nivel de Tarea. De esta forma un programa puede estar asociado a diferentes Tareas y en cada una de ellas definir Valores distintos en los parámetros.



- **Modificar** Tareas.

Para modificar una Tarea se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- Los elementos asociados a la Tarea como: Programa, Cadena, Clases de Tarea y la Planificación Predefinida deben existir previamente.
- **Definición Parámetros** Tarea

Las Tareas asociadas a Programas se podrá definir el *Valor por Defecto* de los parámetros a nivel de Tarea. De esta forma un programa puede estar asociado a diferentes Tareas y en cada una de ellas definir Valores distintos en los parámetros.

- **Eliminar** Tareas.

Para eliminar una Tarea se deben cumplir las siguientes validaciones:

- La tarea no debe estar ejecutándose.

- **Activar** Tareas.

Activa las tareas para que puedan ser ejecutadas.

La primera vez que se Activa una Tarea se comprobara la integridad de todos los elementos asociados a la Tarea.

- **Desactivar** Tareas.

Se podrán Desactivar todas aquellas Tareas que no están en ejecución.

- **Ejecución Inmediata.**

La ejecución inmediata consiste en la ejecución inmediata del Programa o Cadena cadena asociada a la Tarea seleccionada en el momento en que se realice la petición. Para ello la Tarea debe estar Activa.

- **Monitorizar Ejecución.**

Disponible solo para Tareas asociadas a Cadena.

Muestra el estado de ejecución de los pasos en el momento de la ejecución de la Cadena. Las acciones permitidas en esta funcionalidad es la de Completar Paso.

- **Completar Paso.**

Para las tareas en ejecución y están asociadas a una Cadena se les permite modificar el estado de sus pasos para alterar la ejecución.

Para modificar el estado del paso, la cadena en ejecución debe cumplir los siguientes requerimientos:

- El paso no debe estar iniciado
- El estado al que se puede modificar es:
 - NOT STARTED
 - SUCCEEDED
 - FAILED. En este caso hay que indicar el código del error.

- **Parar Tarea.**



Finaliza la ejecución de la tarea en ejecución.

- **Revisar Resultados.**

Muestra el resultado de cada una de ejecuciones de una tarea, incluyendo los resultados de ejecución de cada uno de sus pasos si la Tarea esta asociada a una Cadena.

- **Suscriptores.**

Muestra los Suscriptores asociados a la Tarea.



Módulo Gestión Suscripciones

Este Modulo gestiona las suscripciones de los usuarios a las tareas.

Este nuevo concepto de **Suscripción** viene por la necesidad de informar a un usuario o conjunto de usuarios del resultado obtenido tras la ejecución de una tarea, o facilitar un informe que se ha generado tras la ejecución, a dicho informe se le conoce como **Notificación**.

Este modulo debe ser capaz de:

- Gestionar las suscripciones de los usuarios a una tarea.
- Indicar el modo de envío de las Notificaciones (SMS o Correo Electrónico).
- Personalizar dichas Notificaciones.

Mantenimiento de Suscriptores

Una **Suscripción** hace posible que un usuario reciba una **Notificación** en el que se informa del resultado de la Tarea que se ha ejecutado, así como de los resultados obtenidos en los mismos.

Una Suscripción relaciona un usuario con una Tarea y define cuándo se quiere recibir la notificación de la ejecución de la misma, así como que información se quiere recibir.

La recepción de la Notificación podrá ser:

- Por correo electrónico. Se envía al usuario la información sobre la ejecución de la tarea adjuntando los ficheros de resultado que se hayan definido.
- Por SMS. Se envía al usuario la información sobre la ejecución de la tarea.

Para cada suscripción será necesario indicar que ficheros de los que genera la tarea se desea realizar la suscripción. Para ello se muestra una lista con los ficheros disponibles que se van a generar tras la ejecución de la Tarea, el usuario escogerá de la lista los ficheros que desea suscribirse y en que situación.

La lista de ficheros de la Tarea se obtiene a partir de la acción asociada:

- Si la acción es un Programa: Consultar los Resultados asociados al programa.
- Si la acción es una Cadena: Ver que acción esta asociada a cada paso:
 - Si esta asociado a un programa: Consultar los Resultados asociados al programa.
 - Si esta asociado a un evento: No tiene Resultados.
 - Si esta asociado a una Cadena: Volver a ver las acciones asociadas a cada paso.

El sistema de envío de Notificaciones, por correo electrónico y/o SMS, con el resultado estará basado en el evento de finalización de la Tarea como desencadenante para que se realice el envío.

La Tareas del *Gestor de Suscriptores* son:

- Procesar los eventos de finalización de tareas emitidos por Scheduler.
- Generar las Notificaciones correspondientes con los anexos indicados en la Suscripción.
- Enviar las Notificaciones a los Suscriptores.

En el esquema siguiente se ilustra el modelo de comunicación de eventos entre Scheduler y el *Gestor de Suscripciones*:

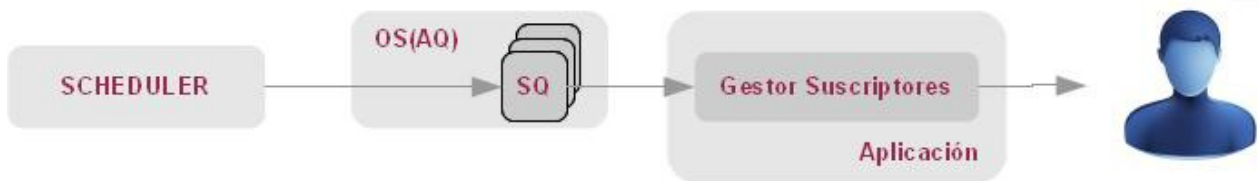


Ilustración 9: Modelo Comunicación Gestor Suscriptores

En la siguiente tabla se describen los eventos que pueden desencadenar un envío de Notificación:

Evento Desencadenantes Envío Notificación	
Estado	Descripción
SUCCEEDED	La Tarea ha finalizado correctamente.
FAILED	La Tarea ha finalizado pero con errores.
STOPPED	La Tarea ha sido parada manualmente.
STALLED	La cadena asociada a la solicitud se encuentra atascada. Precisa de intervención manual para cambiar de estado.
COMPLETED	La Tarea ha finalizado en alguno de los siguientes estados: <ul style="list-style-type: none"> • SUCCEEDED • FAILED • STOPPED

Tabla 13: Eventos Desencadenantes Envío Notificación

En las tablas siguientes se describen:

- Los campos asociados a un Suscriptor junto con las condiciones de obligatoriedad y editabilidad.
- Los datos los Ficheros disponibles para la Suscripción con las condiciones de obligatoriedad y editabilidad.



Datos SUSCRIPTOR			
Campo	Descripción	Oblig	Edit
Tarea	Identificador de la Tarea	S	N
Suscriptor	Identificador de la Suscripción	S	N
Correo	Dirección de correo electrónico al que hay que enviar la Notificación.	N	S
Teléfono	Número de teléfono móvil al que hay que enviar la Notificación.	N	S
Condición	Situación en la que se quiere recibir la Notificación, en función del resultado de la ejecución de la Tarea: <ul style="list-style-type: none"> • COMPLETED. Una vez finalizada. • SUCCEEDED. Si ha finalizado con éxito. • FAILED. Si ha finalizado con error. • STOPPED. Si se ha detenido la ejecución de la tarea. • STALLED. Si la tarea esta atascada. 	S	S
Código Error	Permite especifica los Códigos de error para los que se desea recibir la Notificación cuando el resultado de la ejecución es Failed.	N	S
F. Activación	Fecha a partir de la cual la Suscripción es valida.	S	S
F. Des-activación	Fecha a partir de la cual la Suscripción deja de ser valida.	N	S
Origen Correo	Correo a introducir en el remitente del correo. Si no se especifica se incorporará la dirección del Sistema.	N	S
Asunto Correo	Texto a introducir en el asunto del Correo. Si no se especifica se incorporará un texto estándar.	N	S
Cuerpo Correo	Texto a introducir en el cuerpo del Correo. Si no se especifica se incorporará un texto estándar.	N	S

Tabla 14: Datos Suscriptor

Datos FICHERO SUSCRIPCIONES				
Campo	Descripción	Oblig	Edit	
Tarea	Identificador de la Tarea	S	N	
Id. Suscriptor	Identificador de la Suscripción	S	N	
Id. Fichero	Identificador del Fichero	S	S	
Condición	Situación en la que se quiere recibir el Fichero en función de la condición de Notificación de la Tarea:		S	S
	Condición Notificación	Condición Fichero		
	COMPLETED	SUCCEEDED FAILED STOPPED		
	SUCCEEDED	SUCCEEDED		
	FAILED	FAILED		
	STOPPED	STOPPED		
	STALLED	STALLED		

Tabla 15: Datos Fichero Suscripciones



Descripción Funcional

Este mantenimiento debe permitir:

- **Crear** Suscripción.

Para crear una nueva Suscripción se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- Debe existir la Tarea al que se hace referencia.
- Si la condición de envío es FAILED se debe permitir introducir el Código de Error.
- Debe estar informada el Correo o el Teléfono.
- Si se especifica el Correo:
 - Se puede seleccionar los Ficheros que se desea recibir como adjunto.
 - La condición de envío de los Ficheros debe estar acorde a la condición de la Notificación.
 - Se permite editar los campos Origen, Asunto y Cuerpo.

- **Modificar** Suscripción.

Para modificar una Suscripción se deben cumplir las siguientes validaciones:

- Campos obligatorios informados.
- Debe existir la Tarea al que se hace referencia.
- Si la condición de envío es FAILED se debe permitir introducir el Código de Error.
- Debe estar informada el Correo o el Teléfono.
- Si se especifica el Correo:
 - Se puede seleccionar los Ficheros que se desea recibir como adjunto.
 - La condición de envío de los Ficheros debe estar acorde a la condición de la Notificación.
 - Se permite editar los campos Origen, Asunto y Cuerpo.

- **Eliminar** Suscripción.

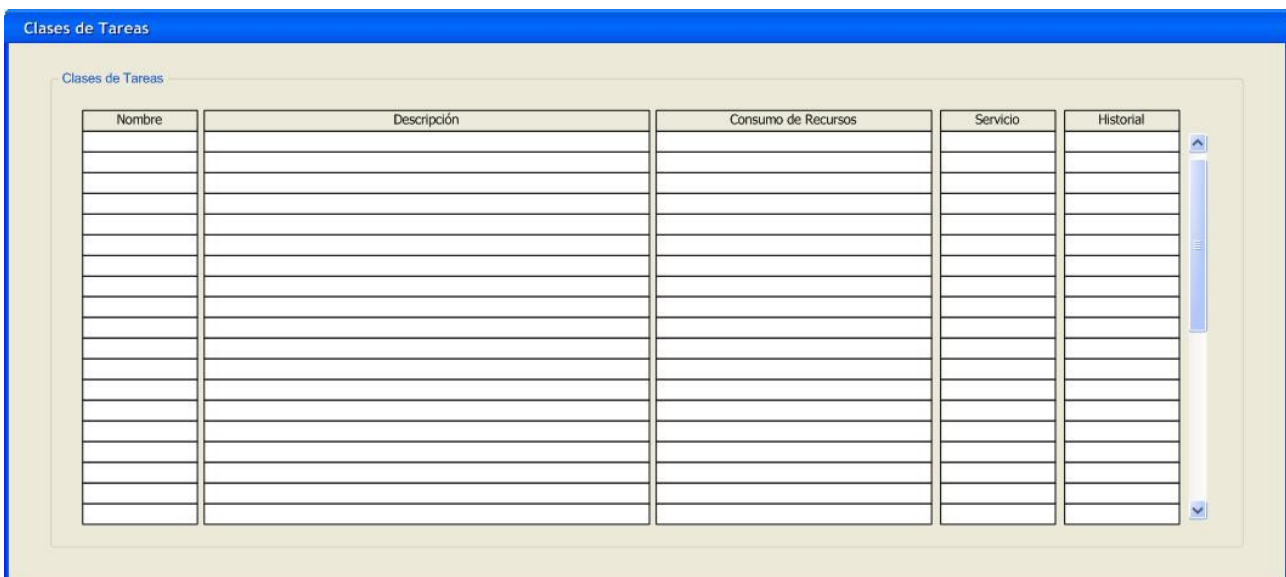
Elimina el Suscriptor de la Tarea.

Diseño Gráfico de la Aplicación

Este apartado describe de forma gráfica una aproximación de como serían los formularios para cada uno de los módulos anteriormente analizados.¹

Mantenimiento Clases de Tarea

Este Mantenimiento muestra en forma de tabla las Clases de Tarea, permitiendo Insertar, Modificar y Eliminar Clases de Tarea.



The screenshot shows a window titled 'Clases de Tareas' with a light blue header bar. Below the header, there is a scrollable area containing a table with the following columns: 'Nombre', 'Descripción', 'Consumo de Recursos', 'Servicio', and 'Historial'. The table has multiple empty rows. A vertical scrollbar is visible on the right side of the table area.

Ilustración 10: Formulario Mantenimiento Clases de Tarea

1 Para el Diseño de los Formularios se ha utilizado la herramienta **Microsoft Visio**.

Mantenimiento Ventanas de Trabajo

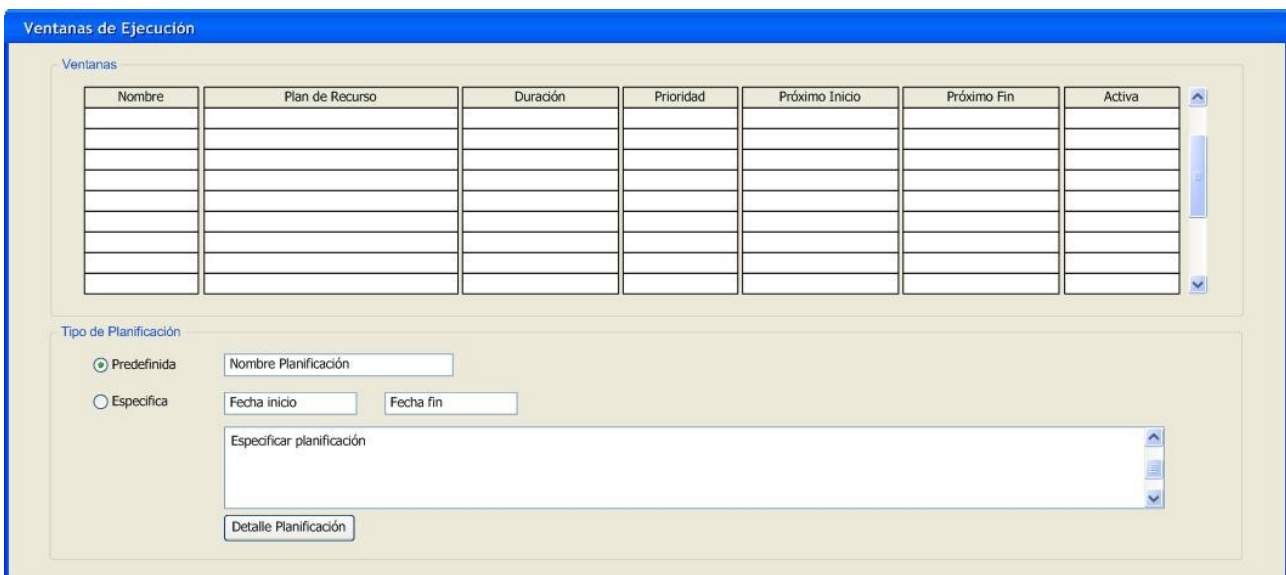
Este Mantenimiento muestra en forma de tabla las Ventanas de Trabajo, permitiendo Insertar, Modificar y Eliminar Ventanas de Trabajo.

A modo de detalle, y para facilitar el mantenimiento, se muestra en la parte inferior de la tabla el “Tipo de Planificación” que se ha establecido para la Ventana de Trabajo.

Las Ventanas de Trabajo permiten dos tipos de Planificaciones:

- Predefinida. Planificación ya existente en Scheduler. Se deberá indicar el Nombre de la planificación en el campo correspondiente, se facilita una lista de valores con las planificaciones ya existentes.
- Específica. Se debe indicar la fecha Inicio y Fin de la Ejecución. Y en el campo correspondiente especificar el Intervalo de Repetición.

Para definir el Intervalo de Repetición se puede hacer uso del formulario “Detalle Planificación” accesible desde el botón con el mismo nombre. El Funcionamiento de dicha ventana se explica en el “Mantenimiento planificaciones”:



Nombre	Plan de Recurso	Duración	Prioridad	Próximo Inicio	Próximo Fin	Activa

Tipo de Planificación

Predefinida

Específica

Ilustración 11: Formulario Mantenimiento Ventanas de Trabajo

Mantenimiento Planificaciones

Este Mantenimiento muestra en forma de tabla las Planificaciones, permitiendo Insertar, Modificar y Eliminar Planificaciones.

A modo de detalle, y para facilitar el mantenimiento, se muestra en la parte inferior de la tabla el bloque “Detalle Planificación”. Este bloque funciona en modo espejo, es decir, muestra la misma información que la tabla de planificaciones pero permitiendo ver todo el contenido del campo.

Para definir el Intervalo de Repetición se puede hacer uso del formulario “Detalle Planificación” accesible desde el botón con el mismo nombre.

El bloque “Evaluar Planificación” evaluará el Intervalo de repetición mostrando las próximas ejecuciones.

Nombre	Fecha Inicio	Fecha Fin	Intervalo Repetición

Detalle Planificación

Nombre Planificación:

Fecha fin:

Intervalo Repetición:

Detalle Planificación

Evaluar Planificación

Próximas Ejecuciones

Evaluar Planificación

Ilustración 12: Formulario Mantenimiento Planificaciones

Scheduler ofrece una extensa sintaxis, que se agrupa mediante cláusulas, para la definición del Intervalo de Repetición que el usuario final no tiene porque saber. Para facilitar el uso de esta sintaxis se ha creado un Formulario de “Detalle Planificación”.

`repeat_interval = frequency_clause [“,” interval_clause] [“,” bymonth_clause] [“,” byweekno_clause] [“,” byweekno_clause] [“,” byyearday_clause] [“,” bydate_clause] [“,” bymonthday_clause] [“,” byday_clause] [“,” byhour_clause] [“,” byminute_clause] [“,” bysecond_clause] [“,” bysetpos_clause] [“,” include_clause] [“,” exclude_clause] [“,” intersect_clause] [“,” periods_clause] [“,” byperiod_clause]`

Este Formulario no recoge todas las cláusulas que ofrece Scheduler para la definición del Intervalo de Repetición, sino que ofrece las cláusulas más utilizadas. Para incluir nuevas cláusulas, se podrá utilizar el campo de texto disponible en el Formulario Principal.

El formulario “Detalle Planificación” se ha creado de forma independiente para que sea accesible desde otros formularios. Por tanto se incluyen los siguientes parámetros de entrada:

- Fecha Inicio
- Fecha Fin
- Intervalo Repetición.

Si se informa alguno de estos parámetros, el formulario deberá rellenar los campos



correspondientes, en el caso del Intervalo de Repetición se deberá desglosar por cláusulas.

En la siguiente tabla se describe todas las cláusulas existentes y una pequeña descripción:

Clausula	Descripción
<i>FREQ</i>	Especifica el tipo de frecuencia. Los posibles valores predefinidos son: YEARLY, MONTHLY, WEEKLY, DAILY, HOURLY, MINUTELY, SECONDLY.
<i>INTERVAL</i>	Especifica el intervalo de repetición de la frecuencia. Su valor por defecto es 1 (Una vez cada segundo en el caso de que la frecuencia sea por segundos) y no puede ser superior a 99.
<i>BYMONTH</i>	Especifica el mes en que se debe de ejecutar. Los meses se pueden definir de forma numérica como de forma abreviada: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
<i>BYWEEKNO</i>	Especifica la semana del año como un número siguiendo la normativa ISO-8601, dichos números son enteros y van desde el 1 hasta el 52 o 53 dependiendo del año. Está normativa es la que se utiliza en los calendarios Gregorianos. Esta cláusula solo estará disponible si <i>FREQ=YEARLY</i>
<i>BYYEARDAY</i>	Especifica los días del año como números que van del 1 hasta el 366.
<i>BYDATE</i>	Especifica las fechas siguiendo el siguiente formato [YYYY]MMDD.
<i>BYMONTHDAY</i>	Especifica los días del mes como números. Valores válidos son del 1 al 31. Se puede utilizar el signo - para contar los días hacia atrás.
<i>BYDAY</i>	Especifica el día de la semana, empezando el Lunes y terminado el Domingo utilizando sus abreviaturas: MON, TUE, WED, THU, FRI, SAT, SUN
<i>BYHOUR</i>	Especifica las Horas (0 - 23)
<i>BYMINUTE</i>	Especifica los Minutos (0 - 59)
<i>BYSECOND</i>	Especifica los Segundos (0 - 59)
<i>BYSETPOS</i>	Especifica la posición de la lista que se obtiene tras evaluar las distintas cláusulas anteriormente definidas.
<i>INCLUDE</i>	Especifica una lista de Planificaciones anteriormente creadas, y las une para crear una nueva.
<i>EXCLUDE</i>	Especifica una lista de Planificaciones anteriormente creadas, y excluye sus ejecuciones de la sintaxis que se ha creado.
<i>INTERSECT</i>	Especifica una lista de Planificaciones anteriormente creadas, y crea una nueva con la intersección de la ejecuciones de dichas planificaciones.
<i>PERIODS</i>	Especifica el número de periodos que debe ejecutarse.
<i>BYPERIOD</i>	Especifica el número de periodo a ejecutar.

Tabla 16: Cláusulas para definir el Intervalo de Repetición

Detalle Planificaciones

Detalle Planificación

Fecha Inicio Fecha Fin

Frecuencia

Intervalo

Meses Ene Feb Mar Abr May Jun Jul Ago Sep Oct Nov Dic

Semana

Fecha

Día de año

Día de mes

Día de semana Lun Mar Mie Jue Vie Sab Dom

Hora

Minutos

Posición

Ilustración 13: Formulario Detalle Planificación



Mantenimiento Eventos

Este Mantenimiento muestra en forma de tabla los Eventos, permitiendo Insertar, Modificar y Eliminar Eventos.

Los Eventos están asociados a una Cola existente en el sistema, se mostrará en el campo de Colas una Lista de Valores con las Colas disponibles.

The screenshot shows a web application interface titled "Eventos Externos". Inside, there is a section labeled "Eventos" containing a table with the following columns: "Evento", "Cola", "Fecha Inicio", "Fecha Fin", "Descripción", and "Condición". The table is currently empty and has a vertical scrollbar on the right side.

Ilustración 14: Formulario Mantenimiento Eventos

Gestor de Programas

El Gestor de programas esta compuesto por:

- Árbol de navegación. Agrupa los programas según a la aplicación a la que pertenece.
- Panel que divide la información del programa en diferentes pestañas.

Al seleccionar un Programa del árbol se deberá recuperar toda la información de dicho programa y completar los campos de las distintas pestañas.

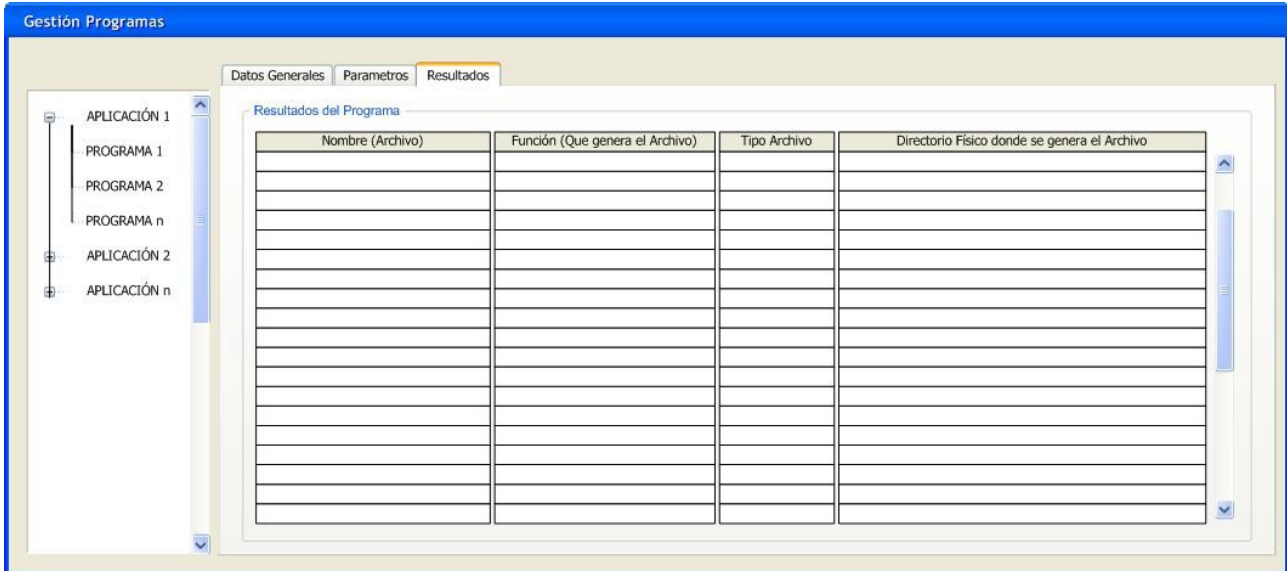
Al realiza una búsqueda utilizando los paneles, se deberá posicionar en el nodo del árbol correspondiente al programa buscado.

The screenshot shows the 'Gestión Programas' interface with the 'Datos Generales' tab active. On the left is a tree view showing a hierarchy of 'APLICACIÓN 1', 'PROGRAMA 1', 'PROGRAMA 2', 'PROGRAMA n', 'APLICACIÓN 2', and 'APLICACIÓN n'. The main panel contains several input fields: 'NOMBRE PROGRAMA', 'DESCRIPCIÓN PROGRAMA', 'APLICACIÓN', 'DESCRIPCIÓN APLICACIÓN (Campo Recuperado)', 'TIPO PROGRAMA' (dropdown), 'ACCIÓN que realiza el Programa', 'USUARIO RESPONSABLE DEL PROGRAMA', 'TIEMPO ESTIMADO DE EJECUCIÓN', and 'NUMERO DE PARÁMETROS'. A 'Crear Cadena Simple' button is at the bottom.

Ilustración 15: Formulario Gestor Programas (Datos Generales)

The screenshot shows the 'Gestión Programas' interface with the 'Parámetros / Argumentos' tab active. The left tree view is the same as in the previous screenshot. The main panel displays a table with four columns: 'Posición', 'Nombre', 'Tipo Dato', and 'Valor por defecto'. The table is currently empty, showing only the header row.

Ilustración 16: Formulario Gestor Programas (Parámetros / Argumentos)



The screenshot shows a software interface titled "Gestión Programas". It has three tabs: "Datos Generales", "Parametros", and "Resultados". The "Resultados" tab is active, displaying a table with the following columns: "Nombre (Archivo)", "Función (Que genera el Archivo)", "Tipo Archivo", and "Directorio Físico donde se genera el Archivo". The table is currently empty. On the left side, there is a tree view with the following items: "APLICACIÓN 1", "PROGRAMA 1", "PROGRAMA 2", "PROGRAMA n", "APLICACIÓN 2", and "APLICACIÓN n".

Ilustración 17: Formulario Gestor Programas (Resultados)

La pestaña Parámetros solo estará habilitada si se especifica el Numero de Parámetros en la Pestaña de Datos Generales. Este campo se rellenará de forma automática si el Programa es de tipo Procedimiento Almacenado.

Gestor de Cadenas

El Gestor de cadenas esta compuesto por:

- Árbol de Navegación. Agrupa las Cadenas según a la aplicación a la que pertenece.
- Panel que divide la información de la Cadena en diferentes pestañas.

Al seleccionar una Cadena del árbol se deberá recuperar toda la información de dicha Cadena y completar los campos de las distintas pestañas.

Al realiza una búsqueda utilizando los paneles, se deberá posicionar en el nodo del árbol correspondiente a la cadena buscada.

Nombre Cadena	Descripción Cadena	Estado	Aplicación	Responsable

Nombre Paso	Tipo Paso	Acción	Ejecutable (Campo Recuperado)

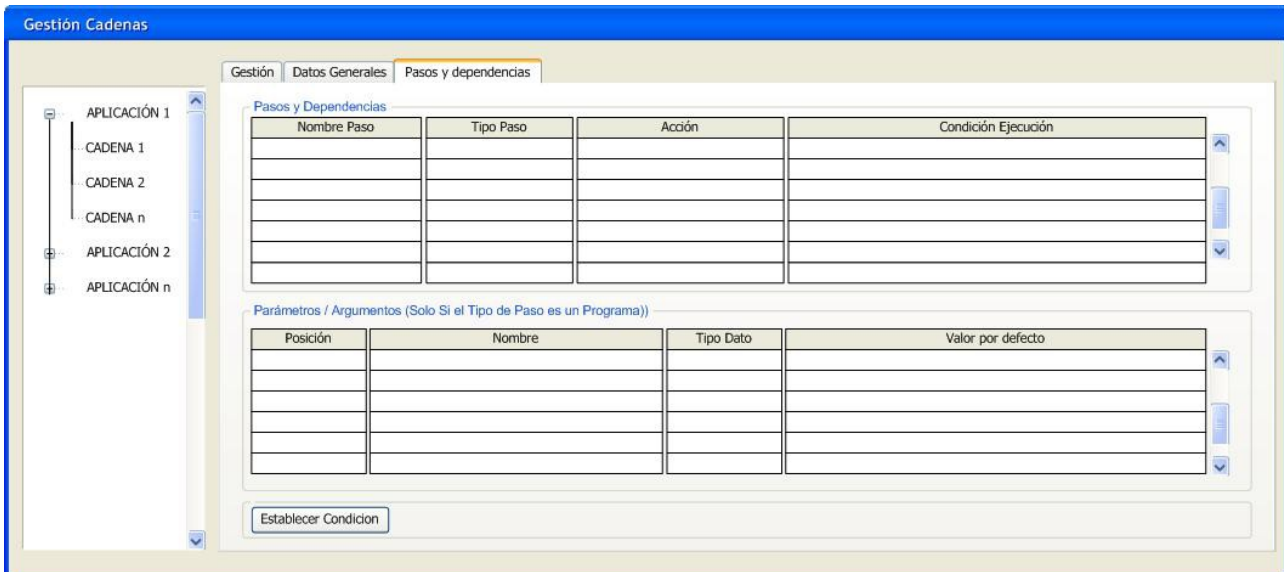
Nombre Paso	Acción

Nombre Paso	Acción

Ilustración 18: Formulario Gestor Cadenas (Gestión)

Acción Finalización	Condición Finalización Cadena

Ilustración 19: Formulario Gestor Cadenas (Datos Generales / Condiciones Finalización)



The screenshot shows the 'Gestión Cadenas' application interface. It has a blue header bar with the title 'Gestión Cadenas'. Below the header, there are three tabs: 'Gestión', 'Datos Generales', and 'Pasos y dependencias'. The 'Pasos y dependencias' tab is active. On the left side, there is a sidebar with a tree view showing 'APLICACIÓN 1' (containing 'CADENA 1', 'CADENA 2', and 'CADENA n') and 'APLICACIÓN 2' (containing 'APLICACIÓN 2' and 'APLICACIÓN n'). The main area contains two tables. The first table, titled 'Pasos y Dependencias', has four columns: 'Nombre Paso', 'Tipo Paso', 'Acción', and 'Condición Ejecución'. The second table, titled 'Parámetros / Argumentos (Solo Si el Tipo de Paso es un Programa)', has four columns: 'Posición', 'Nombre', 'Tipo Dato', and 'Valor por defecto'. At the bottom of the main area, there is a button labeled 'Establecer Condición'.

Ilustración 20: Formulario Gestor Cadenas (Pasos y Dependencias)

El bloque de Parámetros y Argumentos se incluye para modificar el valor por defecto de los parámetros asociados al programa. Es decir, cuando un paso referencia a un programa se recuperarán sus parámetros permitiendo modificar el valor que tomará dicho parámetro a la hora de la ejecución del programa.


La pestaña de Gestión se crea para tener una visión general de la Cadena y de los pasos que contiene. A la hora de Crear o Modificar Cadenas se tiene las otras dos pestañas.

Para crear una Cadena se debe seguir los siguientes pasos:

- Indicar los Datos Generales (Pestaña Datos Generales)
- Indicar los Pasos y Dependencias (Pestaña Pasos y Dependencias)
- Indicar las Condiciones de Fin de ejecución (Pestaña Datos Generales)

Para la definición de las Dependencias entre Pasos y las Condiciones de Fin de Ejecución se puede utilizar el formulario de ayuda "Establecer Condición" disponible desde el botón con el mismo nombre.

La opción "Ejecución Inicial" solo estará disponible cuando se llame desde la pestaña de Pasos y Dependencias. El desplegable de Nombre Paso se deberá rellenar de forma dinámica según los Pasos que se hayan definido para la Cadena actual.



Establecer Condición

Ejecución Inicial

Establecer Condición

Condición NOMBRE PASO EN ESTADO ESTADOS DISPONIBLES Negar Condición

Código Error NOMBRE PASO TIPO ERROR CODIGO ERROR Negar Código Error

OPERADOR LÓGICO Añadir Condición

CONDICIONES INTRODUCIDAS (Se permite editar)

Aceptar Cancelar

Ilustración 21: Formulario Gestor Cadenas (Establecer Condición)

Gestor de Tareas

El Gestor de tareas esta compuesto por:

- Árbol de Navegación. Agrupa las tareas según a la aplicación a la que pertenece.
- Panel que divide la información de la Tarea en diferentes pestañas.

Al seleccionar una tarea del árbol se deberá recuperar toda la información de dicha tarea y completar los campos de las distintas pestañas.

Al realiza una búsqueda utilizando los paneles, se deberá posicionar en el nodo del árbol correspondiente a la tarea buscada.

The screenshot shows the 'Gestión Tareas' application interface. On the left, there is a tree view with nodes for 'APLICACIÓN 1', 'TAREA 1', 'TAREA 2', 'TAREA n', 'APLICACIÓN 2', and 'APLICACIÓN n'. The main area displays a table with the following columns: 'Nombre Tarea', 'Activado', 'Estado', 'Última Ejecución', 'Próxima Ejecución', 'Acción', 'Aplicación', and 'Responsable'. Below the table, there are buttons for 'Activar', 'Desactivar', 'Parar Tarea', 'Ejecución Inmediata', and 'Consultar Acción'.

Ilustración 22: Formulario Gestor Tareas (Gestión)

The screenshot shows the 'Gestión Tareas' application interface with the 'Datos Generales' tab selected. The main area contains several input fields and dropdown menus for configuring a task. Fields include 'NOMBRE TAREA', 'DESCRIPCIÓN TAREA', 'TIPO ACCIÓN', 'NOMBRE ACCIÓN', 'CLASE TAREA', 'ESTADO TAREA', 'PRIORIDAD EJECUCIÓN', 'NÚMERO MÁXIMO ERRORES', 'HISTORIZACIÓN', and 'DESCRIPCIÓN HITORIZACIÓN'. There are also radio buttons for 'Tipo de ejecución' (Inmediata, Evento, Predefinida, Especifica) and input fields for 'NOMBRE EVENTO', 'NOMBRE PLANIFICACIÓN', 'FECHA INICIO', and 'INTERVALO REPETICION'. On the right, there are sections for 'Próximas Ejecuciones' and 'Resumen Ejecuciones' with corresponding input fields. At the bottom, there are buttons for 'Establecer Planificación' and 'Próximas Ejecuciones'.

Ilustración 23: Formulario Gestor Tareas (Datos Generales)

Posición	Nombre	Tipo Dato	Valor por defecto

Ilustración 24: Formulario Gestor Tareas (Parámetros)

Nombre Cadena	Nombre Paso	Acción del Paso	Estado Paso	Código Error	Fecha Inicio	Fecha Fin

Modificar Estado Parar Tarea

Ilustración 25: Formulario Gestor Tareas (Monitorizar Ejecución)

La pestaña de Gestión esta sincronizada con el Árbol, es decir al seleccionar una Tarea del Árbol se seleccionará la Tarea en la Tabla. Y al seleccionar una Tarea de la Tabla se posicionará en el nodo de la Tarea correspondiente.

Desde esta pestaña se puede Activar y Desactivar Tareas. Parar la ejecución de una tarea. Ejecutar de forma inmediata una tarea. Consultar el Programa o Cadena asociada a la Tarea (Navegará al Formulario de Gestión Programa o Cadena en modo consulta).

La pestaña de Parámetros estará habilitada solo cuando la Tarea especifica a un Programa. En Tal caso se podrá modificar el valor por defecto de los parámetros asociados al programa para cuando se ejecute la Tarea.

La pestaña Monitorizar Ejecución solo estará habilitada cuando la Tarea referencia a una Cadena y dicha Tarea esta ejecutándose, permitiendo modificar el estado de cualquier paso que no se haya ejecutado



aún, o parar la ejecución de la tarea.

Para crear y modificar las tareas ya existentes se utilizará la pestaña de Datos Generales. El campo Nombre Acción debe mostrar una lista de valores dependiendo del tipo Acción seleccionado (Programa o Cadena), El campo Clase Tarea debe mostrar una lista de valores con las Clases ya creadas.

Se permiten cuatro tipos de ejecuciones diferentes:

- Ejecución Inmediata
- Evento: Indicando el Nombre del Evento. Se facilita lista de valores con los eventos creados.
- Planificación predefinida: indicando el Nombre de la planificación. Se facilita lista de valores con las planificaciones creadas.
- Planificación Específica: se define una planificación exclusiva para la Tarea. Para definir el Intervalo de repetición se puede utilizar el formulario de "*Detalle Planificación*"

En dicha pestaña se puede ver las próximas ejecuciones, y a modo de estadística se tiene el bloque de Resumen de ejecución, donde se informa del Total de ejecuciones y cuantas han sido Correctas, Incorrectas y cuantas se han parado.

Gestor de Suscriptores

El Gestor de Suscriptores esta compuesto por un panel que contiene dos pestañas

- Suscripciones: Muestra solamente aquellas tareas en que el usuario esta suscrito.
- Todas: Muestra todas las tareas y para cada una de ellas, que usuarios esta suscrito.

Cada Pestaña tiene su Árbol de Navegación. Al seleccionar una tarea del árbol se deberá recuperar toda la información de dicha tarea. Al realizar una búsqueda, se deberá posicionar en el nodo del árbol correspondiente a la tarea buscada.

La peculiaridad de este Gestor es el bloque de Selección de Ficheros. Este bloque muestra todos los Resultados correspondiente al programa asociado a la Tarea. En el caso de que la Tarea este asociado a una Cadena, muestra todos los resultados asociados a los Resultados de los programas que ejecutan los pasos de dicha cadena.

La condición de suscripción del resultado dependerá de la condición de suscripción que se haya seleccionado en la tarea.

Correo	Movil	Nombre	Apellido 1	Apellido 2	Condición	Error

Nombre Tarea	Nombre Acción	Aplicación	Descripción	Activa	Desactiva

Nombre Fichero / Función	Condición

Ilustración 26: Formulario Gestor Suscripciones (Suscripciones)

Nombre Tarea	Nombre Acción	Aplicación	Descripción	Activa	Desactiva

Correo	Movil	Nombre	Apellido 1	Apellido 2	Condición	Error

Nombre Fichero / Función	Condición

Ilustración 27: Formulario Gestor Suscripciones (Todas)



4 Disección de la aplicación

En este apartado se va a diseccionar la aplicación, para ello se va a utilizar un lenguaje de Modelado Unificado (**UML** – *Unified Modeling language*), es decir, un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de esta aplicación.²

Para ello se va a crear los siguientes modelos:

- Diagrama de Clases
- Diagrama Entidad Relación

4.1 Diagrama de Clases

Conceptos básicos

Un diagrama de clase sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento.

Un diagrama de clase esta compuesto por los siguientes elementos:

- Clase : Atributos, Métodos y Visibilidad
- Relación : Herencia, Composición, Agregación y Uso

Elementos

La **Clase** es la unidad básica que encapsula toda la información de un objeto (un objeto es una instancia de una clase). A través de ellas se puede modelar el entorno de estudio.

En UML, una clase esta representada por un rectángulo que posee tres divisiones:

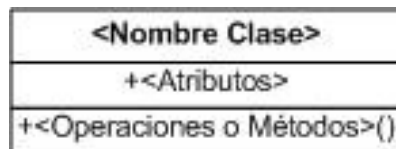


Ilustración 28: Diagrama Clases. Conceptos Básicos

En donde:

- Superior : Contiene el nombre de la clase.
- Intermedio : Contiene los atributos (o variables de instancia) que caracterizan a la clase (pueden ser private, protected o public).
- Inferior : Contiene los métodos u operaciones, los cuales son la forma como intercara el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).

Los **Atributos** o características de una clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno, estos son:

- (+) Public : Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos los lados.
- (-) Private : Indica que el atributo solo será accesible desde dentro de la clase (solo sus métodos lo pueden utilizar)
- (#) Protected : Indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser

2 Para el Diseño de los Diagramas se ha utilizado la herramienta **Microsoft Visio**.

utilizado por métodos de la clase además de las subclases que se deriven.

Los **Métodos** u Operaciones de una clase son la forma en como está intercara con su entorno, éstos pueden tener las siguientes características:

- (+) **Public** : Indica que el método será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos los lados.
- (-) **Private** : Indica que el método solo será accesible desde dentro de la clase (solo otros métodos lo pueden utilizar)
- (#) **Protected** : Indica que el método no será accesible desde fuera de la clase, pero si podrá ser utilizado por métodos de la clase además de las subclases que se deriven.

Relaciones entre clases

Ahora que ya se ha definido el concepto de clase, es necesario explicar como se pueden interrelacionar dos o más clases (cada uno con sus características y objetos). Pero antes es necesario explicar el concepto de *cardinalidad* en relaciones.

En UML, la **cardinalidad** de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y estas pueden ser:

- uno a muchos : (1..n) (1..*)
- cero a muchos : (0..n) (0..*)
- número fijo : m (m denota un número)

Los distintos tipos de relaciones que existen son:

- Herencia (Especialización / Generalización)

Indica que una subclase hereda los métodos y atributos especificados por una SuperClase, por ende la SubClase además de poseer sus propios Métodos y Atributos, poseerá las características y atributos visibles de la SuperClase (Public o Protected)

- Agregación

Para modelar objetos complejos, objetos que son instancias de clase definidas por el desarrollador de la aplicación, tenemos dos posibilidades:

- **Por Valor** : Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido esta condicionado por el tiempo de vida del que lo incluye. Este tipo de relación es común mente llamada Composición (El objeto base se construye a partir del objeto incluido, es decir, es "parte/todo").
- **por Referencia** : Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada Agregación (El objeto base utiliza al incluido para su funcionamiento).

- Asociación

Esta relación permite asociar objetos que colaboran entre sí. Cabe destacar que no es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.

- Dependencia o Instanciación (uso)

Relación bastante particular en que una clase es instanciada (su instanciación es dependiente de otro objeto/clase).

- Casos Particulares



- El objeto de la clase scheduler_schedules puede hacer referencia en alguno objeto de las clases: scheduler_windows, scheduler_chains o scheduler_jobs
- El objeto de la clase scheduler_chain precisa de una referencia a scheduler_schedules o scheduler_programs.
- El objeto de la clase scheduler_jobs precisa de una referencia a scheduler_windows o scheduler_windows_group, precisa referencias a un objeto de scheduler_job_classes y precisa una acción que puede ser un objeto de la clase scheduler_chain o scheduler_programs
- Existen también varias relaciones de composición, una clase referencia a otra de forma dependiente, es decir cuando el objeto de la clase referenciada es eliminada, los objetos que dependen de ella también se eliminan.
 - Las clases scheduler_chain_steps y scheduler_chain_rules precisan de un objeto scheduler_chains para poder ser creados.
 - Las clases scheduler_programs_args y scheduler_program_result precisan de un objeto scheduler_programs para poder ser creados
 - Las clases scheduler_job_args y scheduler_job_suscriber precisan de un objeto scheduler_jobs para poder ser creados.

4.2 Esquema Base de Datos

El **esquema de una base de datos** (*database Schema*) describe la estructura de la base de datos, en un lenguaje formal soportado por un sistema de gestión de base de datos (DBMS). En una base de datos relacional, el esquema define las tablas, los campos asociados a cada tabla y las relaciones entre cada campo y cada tabla.

Un **Modelo de Datos** se utiliza para diseñar la estructura de una Base de Datos mediante una colección de herramientas conceptuales que permiten describir los datos, las entidades que intervienen, las relaciones entre las diferentes entidades u objetos y las restricciones de consistencia.

La base de datos de Oracle es relacional por tanto el modelo de datos a utilizar va a ser el **Modelo Entidad-Relación**.

Modelo Entidad-Relación

Como ya se ha comentado este modelo es solo y exclusivamente un método de que disponemos para diseñar esquemas que posteriormente deberán ser implementados en un gestor de BBDD (Base de datos). Este modelo se representa a través de diagramas y esta formado por varios elementos.

Este modelo habitualmente, además de disponer de un diagrama que ayuda a entender los datos y como se relacionan entre ellos, debe ser completado con un pequeño resumen con la lista de los atributos y las relaciones de cada elemento.

Los elementos del Modelo Entidad-Relación son:

- **Entidad** : Representan cosas u objetos (ya sean reales o abstractos), que se diferencian claramente entre si.
- **Atributos** : Definen o identifican las características de la entidad (es el contenido de la entidad). Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha...)
- **Relación** : Se trata de un vínculo que permite definir una dependencia entre varias entidades, es decir, permite exigir que varias entidades compartan ciertos atributos de forma indispensable.

El concepto de **cardinalidad** vuelve estar presente para definir los distintos tipos de relaciones entre las entidades:

- **Uno a Uno** : Determina que un registro de una entidad solo esta relacionada con un registro de otra entidad y viceversa.
- **Uno a Varios o Varios a Uno**: Determina que un registro de una entidad puede estar relacionado con varios de otra entidad.
- **Varios a Varios** : Determina que una entidad puede relacionarse con otra con ninguno o varios registros y viceversa.

Otro concepto a tener en cuenta en estos Modelos son las **Claves**. Las claves son restricciones que se aplican a un atributo para distinguir unos registros de otros dentro de la misma entidad. Existen varios tipos de clave:

- **Clave Primaria** : Se trata de una clave que define de forma unívoca a toda la Entidad. Los atributos que componen la Clave primaria deben ser únicos, es decir, no se pueden repetir.
- **Clave Foránea** : Se trata de una referencia a una clave primaria de otra Entidad determinando de esta forma la relación existente entre las dos Entidades. Los atributos que componen esta clave no es necesario que pertenezcan a la Clave Primaria de la Entidad.

- Clave Índice: Estas claves surgen por la necesidad de tener un acceso más rápido a los datos de la Entidad. Pueden ser creados con cualquier combinación de los atributos de la Entidad.

Diagrama E-R del Modelo de Datos

El siguiente diagrama muestra el Modelo Entidad-Relación para la aplicación Scheduler. Mencionar que en dicho diseño no se han incluido en las Entidades (tablas) todos los campos disponibles, ya que la mayoría de ellos no se utilizan, al menos por lo que respecta al diseño de esta aplicación.

En este diagrama tampoco se ha incluido todas las Entidades (tablas) que posee Scheduler, ya que como se ha mencionado antes son irrelevantes para la aplicación.

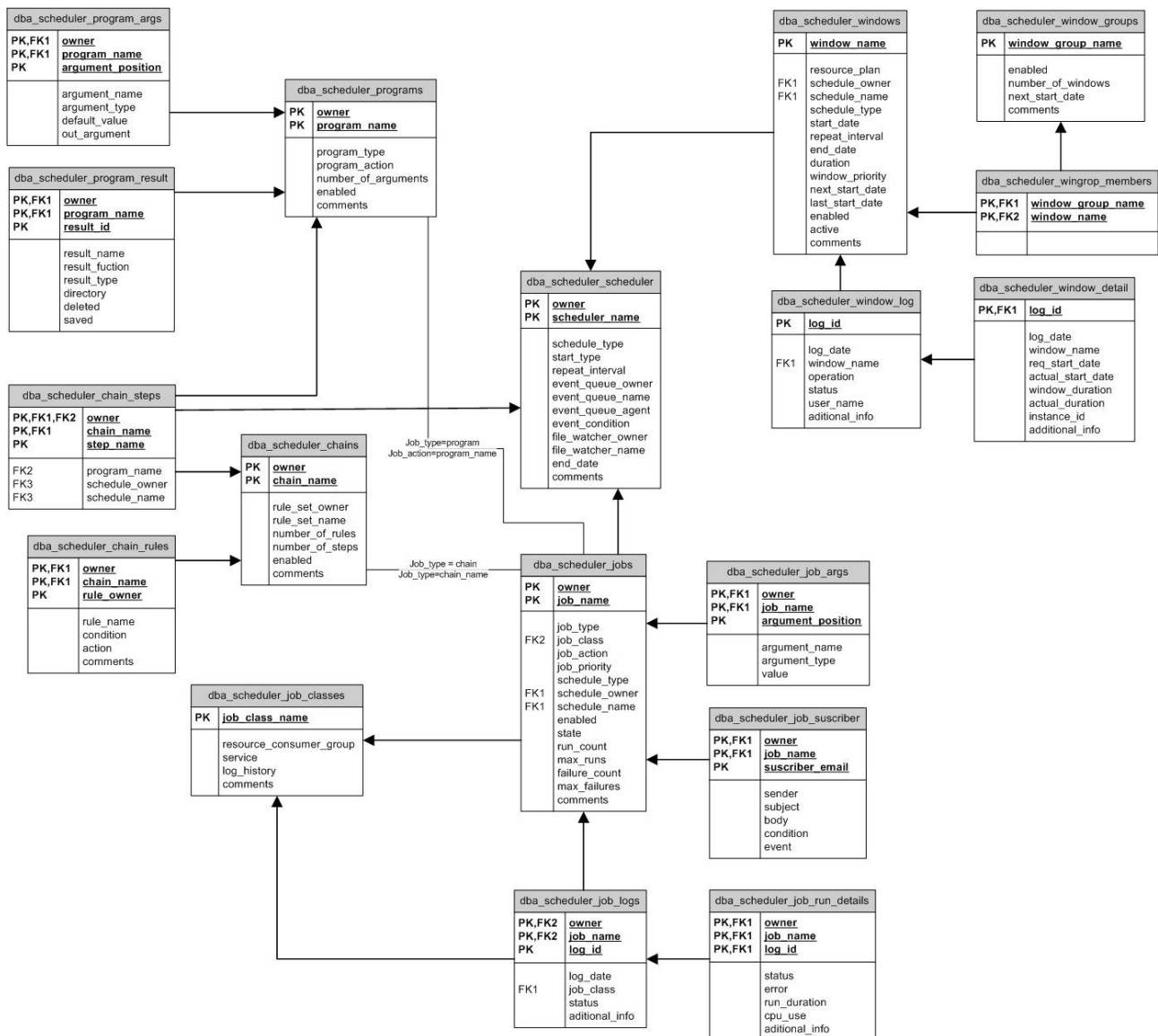


Ilustración 30: Diagrama ER. Scheduler

5 Organización del código

El código se organiza teniendo en cuenta el **Modelo Vista Controlador (MVC)**. Este modelo es un patrón de arquitectura de software, que separa los datos y la lógica de negocio, de la interfaz de usuario y el módulo que gestiona eventos y comunicaciones.

El modelo MVC implica la construcción de tres componentes distintos que son el **modelo**, la **vista** y el **controlador**, es decir por un lado se define los componentes para la representación de la información, y por otro lado la interacción del usuario.

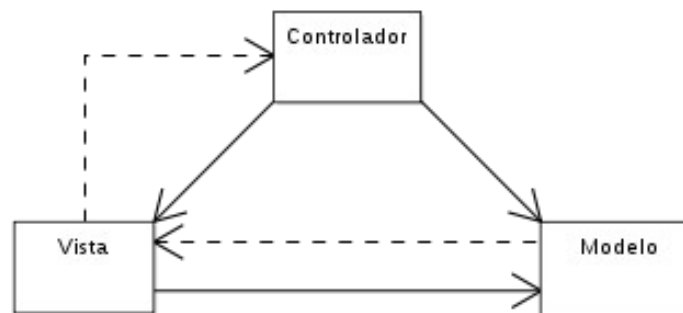


Ilustración 31: Modelo Vista Controlador (MVC)

De manera genérica, los componentes MVC se podrían definir como:

- **Modelo** : Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'
- **Controlador** : Responde a eventos (usualmente acciones de usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información. También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo', por tanto, se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.
- **Vista** : Presenta el 'modelo' en un formato adecuado para interactuar (conocido usualmente como interfaz de usuario).

La aplicación que se está analizando, utiliza un *Sistema de Gestión de Base de Datos*; en líneas generales del **MVC** dicha gestión corresponde al **Modelo**. La unión entre *capa de presentación* y *capa de negocio*, conocido como el paradigma de la programación por capas, representaría la integración entre la **Vista** y su correspondiente **Controlador** de eventos y acceso a datos. MVC no pretende discriminar entre capa de negocio y capa de presentación pero sí pretende separar la *capa visual gráfica* de su correspondiente *programación* y *acceso a datos*, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo.

Atendiendo a lo anteriormente expuesto este punto se dividirá en dos bloques:

- Implementación Base de Datos : En este bloque se realiza una introducción a la programación SQL y PL/SQL, para entender la organización del código de cada una de las clases.
- Implementación Vista – Controlador : En este bloque se realiza una introducción al desarrollo de formularios utilizando la herramienta **Oracle Forms Developer**.



5.1 Implementación Modelo

El Modelo se implementa en la base de datos, pero antes de explicar como se organiza el código es necesario realizar una introducción al lenguaje PL/SQL para entender mejor lo que se va a realizar.

Introducción PL/SQL

SQL es un lenguaje de consulta para los sistemas de base de datos relacionales, pero que no posee la potencia de los lenguajes de programación.

PL/SQL amplía SQL con los elementos característicos de los lenguajes de programación, variables, sentencias de control de flujo, bucles ...

Cuando se desea realizar una aplicación completa para el manejo de una base de datos relacional, resulta necesario utilizar alguna herramienta que soporte la capacidad de consulta del SQL y la versatilidad de los lenguajes de programación tradicionales. PL/SQL es el lenguaje de programación que proporciona **Oracle** para extender el SQL estándar con otro tipo de instrucciones.

Con PL/SQL se puede programar unidades de programa en la base de datos de **Oracle**, estas son:

- Procedimientos almacenados.
- Funciones.
- Triggers.
- Scripts.

Además PL/SQL permite realizar programas sobre las siguientes herramientas de Oracle:

- Oracle Forms
- Oracle Reports
- Oracle Graphics
- Oracle Application Server

PL/SQL no es CASE-SENSITIVE, es decir, no diferencia mayúsculas de minúsculas como otros lenguajes de programación como C o Java. Sin embargo recordar que Oracle es CASE-SENSITIVE en la búsqueda de texto.

Una línea de PL/SQL contiene grupos de caracteres conocidos como UNIDADES LEXICAS, que pueden ser clasificadas como:

- Delimitadores
- Identificadores
- Literales
- Comentarios
- Expresiones

PL/SQL proporciona una variedad predefinida de tipos de datos. Casi todos los tipos de datos soportados por PL/SQL son similares a los soportados por SQL.

PL/SQL permite agrupar Variables, Cursores, Procedimientos almacenados y Funciones en un paquete. Este objeto de base de datos define de forma sencilla y clara la interfaz de un conjunto de procedimientos y tipos de relaciones que se pueden acceder por sentencias SQL.

Estos paquetes por lo general tiene dos partes: una especificación y un cuerpo. La especificación define la interfaz de programación declarando las constantes, variables, excepciones, cursores,

procedimientos y funciones. El cuerpo contiene el código PL/SQL de los objetos creados en la especificación.

La estructura PL/SQL para crear los procedimientos, funciones y paquetes son los siguientes:

Procedimiento Almacenado
<pre>CREATE [OR REPLACE] PROCEDURE nombre_procedimiento [(parámetro [, parámetro])] IS -- Parte declarativa BEGIN -- Parte de ejecución [EXCEPTION] -- Parte de excepciones END [nombre_procedimiento];</pre>

Tabla 17: Estructura Procedimiento Almacenado

Función
<pre>CREATE [OR REPLACE] FUNCTION nombre_función [(parámetro [, parámetro])] RETURN Data Type IS AS -- Parte declarativa BEGIN -- Parte de ejecución [EXCEPTION] -- Parte de excepciones END [nombre_función];</pre>

Tabla 18: Estructura Función

Cuando se crea un procedimiento o una función, se pueden definir parámetros. Existen tres tipos de parámetros que se pueden declarar:

- **IN** : Parámetro de entrada, no se puede sobrescribir.
- **OUT** : Parámetro de salida, se puede escribir en él.
- **IN OUT** : Parámetro de entrada y salida, se permite sobrescribir.

Paquete
<pre>CREATE [OR REPLACE] PACKAGE nombre_paquete AS -- Especificación del paquete END [nombre_paquete]; / CREATE [OR REPLACE] PACKAGE BODY nombre_paquete AS -- Cuerpo del paquete END [nombre_paquete];</pre>

Tabla 19: Estructura Paquete

Teniendo en mente el Diagrama de Clase de la aplicación y los conceptos introducidos se decide crear un paquete de base de datos para cada una de las clases definida.

Scheduler Común

SCHEDULER_COMUN
+SCHEDULER_OBJECT
+SCHEDULER_ATTRIBUTE
+DISABLE()
+ENABLED()
+GET_ATTRIBUTE()
+SET_ATTRIBUTE()
+SET_ATTRIBUTE_NULL()
+EVALUATE_CALENDAR_STRING()

Ilustración 32: Scheduler Común

Este paquete de base de datos contendrá todos los procesos comunes a los objetos scheduler. La mayoría de estos procesos reciben como parámetro el Nombre del Objeto Scheduler.

Este paquete se organiza como se indica en la siguiente tabla:

Tabla 20: *pck_scheduler_comun*

PCK_SCHEDULER_COMUN		
Paquete que engloba los procesos comunes a la mayoría de Objetos Scheduler.		
DISABLE		
Deshabilita Objetos Scheduler		
Parámetro	in / out	Descripción
name	in	Nombre del Objeto Scheduler
force	in	Deshabilita todos los Objetos Scheduler que dependen del objeto que se esta Deshabilitando. Su valor por defecto es: <i>TRUE</i> .
commit_semantics	in	Soporta los siguientes valores: <ul style="list-style-type: none"> <i>STOP_ON_FIRST_ERROR</i>: Valor por Defecto. Detiene la ejecución tras el primer error detectado. <i>TRANSACTIONAL</i>: Detiene la ejecución tras el primer error detectado. Vuelve a Habilitar los objetos que se han Deshabilitado. <i>ABSORB_ERRORS</i>: No se detiene cuando se encuentra con errores. Deshabilita todos los objetos.
ENABLED		
Habilita Objetos Scheduler		
Parámetro	in / out	Descripción
name	in	Nombre del Objeto Scheduler
commit_semantics	in	Soporta los siguientes valores:



		<ul style="list-style-type: none">• STOP_ON_FIRST_ERROR: Valor por Defecto. Detiene la ejecución tras el primer error detectado.• TRANSACTIONAL: Detiene la ejecución tras el primer error detectado. Vuelve a Deshabilitar los objetos que se han Habilitado.• ABSORB_ERRORS: No se detiene cuando se encuentra con errores. Habilita todos los objetos.
GET_ATTRIBUTE		
Devuelve el valor del atributo asociado al Objeto Scheduler		
Parámetro	in / out	Descripción
name	in	Nombre del Objeto Scheduler
attribute	in	Nombre del atributo (ver tabla atributos por objeto)
value	out	Valor del atributo
SET_ATTRIBUTE		
Cambia el valor del atributo asociado al Objeto Scheduler		
Parámetro	in / out	Descripción
name	in	Nombre del Objeto Scheduler
attribute	in	Nombre del atributo (ver tabla atributos por objeto)
value	in	Nuevo valor del atributo
SET_ATTRIBUTE_NULL		
Cambia el valor del atributo asociado al Objeto Scheduler a NULL (vacío)		
Parámetro	in / out	Descripción
name	in	Nombre del Objeto Scheduler
attribute	in	Nombre del atributo (ver tabla atributos por objeto)
EVALUATE_CALENDAR_STRING		
Evaluá la expresión del calendario (intervalo de repetición) retornando la próxima fecha de ejecución.		
Parámetro	in / out	Descripción
calendar_string	in	Expresión a evaluar.
start_date	in	Fecha a partir de la cual la expresión del calendario será activa. Su valor por defecto es la fecha actual
return_date_after	in	Fecha a partir de la cual se quiere que se retorne la próxima fecha de ejecución. No puede ser menor que la fecha que se indica en el parámetro <i>start_date</i> . Su valor por defecto es la fecha indicada en <i>start_date</i> .
next_run_date	out	Próxima fecha de ejecución a partir de la fecha que se indica en el parámetro <i>return_date_after</i> .

En las siguientes tablas se muestran los atributos que se pueden modificar para cada uno de los Objetos Scheduler.

WINDOW ATTRIBUTE VALUES	
Nombre Atributo	Descripción
comments	Descripción de la ventana
duration	Duración de la ventana
resource_plan	Consumos asociados a la Ventana
window_priority	Prioridad de la ventana (Cuando se superpone con otras.) Sus valores posibles son: <ul style="list-style-type: none"> • <i>LOW</i> (valor por defecto) • <i>HIGH</i>
schedule_name	Nombre del Objeto Scheduler Schedule asociado a la ventana. Atributo es Excluyente de: <i>start_date</i> , <i>end_date</i> y <i>repeat_interval</i>
start_date	Fecha a partir de la cual la Ventana esta abierta. Atributo es Excluyente de: <i>schedule_name</i> Atributo Asociado a: <i>repeat_interval</i>
end_date	Fecha a partir de la cual la ventana deja de ser valida. Atributo es Excluyente de: <i>schedule_name</i> Atributo Asociado a: <i>repeat_interval</i>
repeat_interval	Especifica la expresión del calendario que indica cuando se abrirá la ventana. Atributo es Excluyente de: <i>schedule_name</i>

Tabla 21: atributos ventanas (*windows attribute*)

SCHEDULE ATTRIBUTE VALUES	
Nombre Atributo	Descripción
comments	Descripción de la planificación
event_spec	Especificación del evento.
repeat_interval	Especifica la expresión del calendario que indica cuando se abrirá la ventana.
start_date	Fecha a partir de la cual la planificación estará activa.
end_date	Fecha a partir de la cual la planificación estará desactiva.

Tabla 22: atributos planificaciones (*schedule attribute*)

PROGRAM ATTRIBUTE VALUES	
Nombre Atributo	Descripción
comments	Descripción del programa
program_type	Tipo de programa: <i>PLSQL_BLOCK</i> , <i>STORED_PROCEDURE</i> , <i>EXECUTABLE</i>
program_action	Acción que realiza el programa. Dependerá del tipo de programa que se indique.
number_of_arguments	Número de parámetros que precisa los programas de tipo <i>STORED_PROCEDURE</i> y <i>EXECUTABLE</i> para su correcta ejecución.

Tabla 23: atributos programa (*program attribute*)



CHAIN ATTRIBUTE VALUES	
Nombre Atributo	Descripción
comments	Descripción de la cadena

Tabla 24: atributos cadenas (chain attribute)

JOB CLASS ATTRIBUTE VALUES	
Nombre Atributo	Descripción
comments	Descripción de la clase de tarea
log_history	Indica el numero de días que se almacena los registros. Su valor va desde el 0 al 1.000.000. 0: Indica que no se generar registros
logging_level	Especifica el nivel del registro. Su valor puede ser: <ul style="list-style-type: none">• LOGGING_OFF: No se escribe en el registro.• LOGGING_FAILED_RUNS: Se escriben en el registros solo los errores de la ejecución de la tarea.• LOGGING_RUNS: Se escribe en el registro el detalle de la ejecución de la tarea.• LOGGINF_FULL: Se escribe en el registro todas las operaciones asociados a la clase de la tarea.
resource_consumer_group	Grupo de recursos de sistema que la clase de tarea esta asociada.
service	Servicio de base de datos que la clase de tarea esta asociada.

Tabla 25: atributos clase de tareas (job class attribute)

SUSCRIBER ATTRIBUTE VALUES	
Nombre Atributo	Descripción
condition	Condición de envió de notificación. Depende del resultado que quede la Tarea tras su ejecución: <i>COMPLETED</i> , <i>SUCCEDED</i> , <i>FAILED</i> , <i>STOPPED</i> , <i>STALLED</i> .
error	Si la condición = <i>FAILED</i> . Códigos de Error para la cual se quiere recibir la notificación
start_date	Fecha a partir de la cual la Suscripción esta activa.
end_date	Fecha a partir de la cual la Suscripción esta desactiva.
sender	Correo Origen de la Notificación.
subject	Texto del Asunto de la Notificación.
body	Texto del Cuerpo de la Notificación

Tabla 26: atributos suscripciones (subscriber attribute)



JOB ATTRIBUTE VALUES	
Nombre Atributo	Descripción
comments	Descripción de la Tarea
job_type	Tipo de Objeto reverenciado: Programa o Cadena.
job_action	Nombre del Objeto Scheduler al que hace referencia. Depende del valor que se haya indicado en el atributo: <i>job_type</i>
job_class	Clase de Tarea asociado.
job_priority	Prioridad sobre otras tareas que pertenezcan a la misma clase. Su valor va desde el 1 (prioridad alta) hasta el 5 (prioridad baja). El valor por defecto es el 3.
logging_level	Especifica el nivel del registro. Su valor puede ser: <ul style="list-style-type: none">• LOGGING_OFF: No se escribe en el registro.• LOGGING_FAILED_RUNS: Se escriben en el registros solo los errores de la ejecución de la tarea.• LOGGING_RUNS: Se escribe en el registro el detalle de la ejecución de la tarea.• LOGGING_FULL: Se escribe en el registro todas las operaciones asociados a la clase de la tarea.
max_failures	Numero de ejecuciones fallidas seguidas tras la cual se deshabilitara la tarea.
max_run_duration	Tiempo máximo que puede durar la ejecución. Una vez sobrepasado dicho tiempo salta la excepción JOB_OVER_MAX_DUR. Y el estado de la tarea se queda como FAILED.
max_runs	Número de veces consecutivas que puede ejecutarse la tarea.
schedule_name	Indica el Evento o Planificación asociada a la Tarea para su ejecución. Excluyente de los Atributos: <i>start_date</i> , <i>end_date</i> , <i>repeat_interval</i> .
start_date	Fecha a partir de la cual la cual el intervalo de repetición esta activa. Excluyente del Atributo: <i>schedule_name</i> .
end_date	Fecha a partir de la cual la cual el intervalo de repetición esta desactiva. Excluyente del Atributo: <i>schedule_name</i> .
repeat_interval	Especifica la expresión del calendario que indica cuando se ejecutara la tarea. Excluyente del Atributo: <i>schedule_name</i> .

Tabla 27: atributos tarea (*job attribute*)

Scheduler Windows

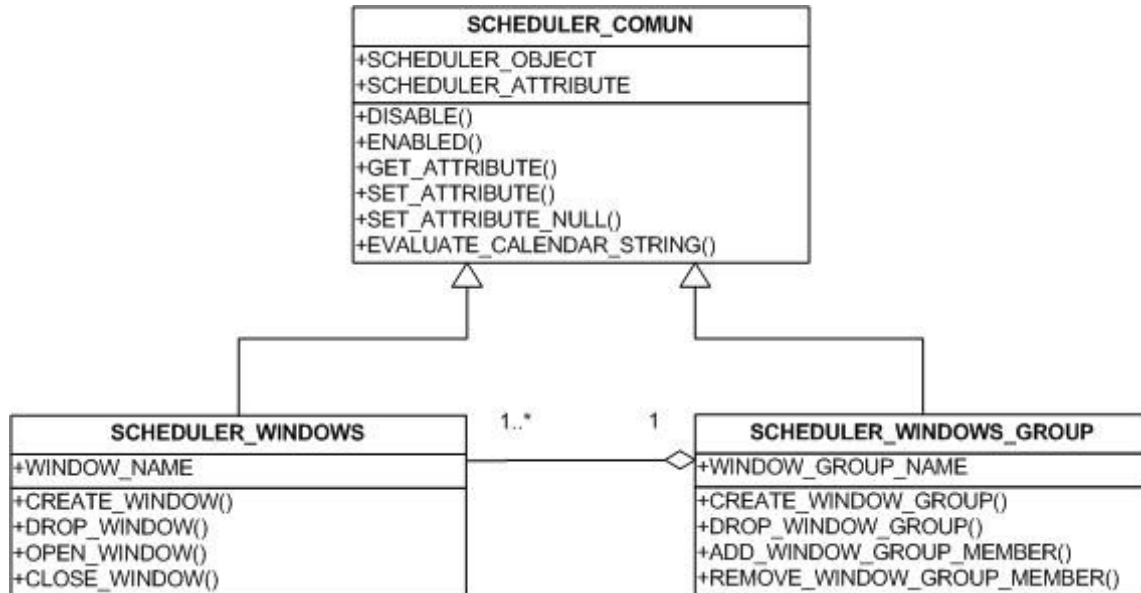


Ilustración 33: Scheduler Windows

Se construye dos paquetes de base de datos que contendrán los procesos específicos para el Objeto de Scheduler que esta asociado:

Tabla 28: *pck_scheduler_windows*

PCK_SCHEDULER_WINDOWS		
Paquete que contiene los procesos específicos a la ventana de trabajo.		
CREATE_WINDOW_GROUP		
Crea una ventana de trabajo que esta asociada a una planificación		
Parámetro	in / out	Descripción
window_name	in	Nombre de la ventana de trabajo
resource_plan	in	Recursos asociados a la ventana de trabajo
schedule_name	in	Nombre de la planificación asociada a la ventana de trabajo.
duration	in	Duración de la ventana de trabajo
window_priority	in	Prioridad de la ventana de trabajo
comments	in	Descripción de la ventana de trabajo
CREATE_WINDOW		
Crea una ventana de trabajo que esta asociada a una planificación específica		
Parámetro	in / out	Descripción
window_name	in	Nombre de la ventana de trabajo
resource_plan	in	Recursos asociados a la ventana de trabajo



start_date	in	Fecha a partir de la cual la ventana de trabajo esta activa.
repeat_interval	in	Específica la expresión del calendario que indica cuando se abre la ventana de trabajo.
end_date	in	Fecha a partir de la cual la ventana de trabajo esta desactiva.
duration	in	Duración de la ventana de trabajo
window_priority	in	Prioridad de la ventana de trabajo
comments	in	Descripción de la ventana de trabajo
DROP_WINDOW		
Elimina la ventana de trabajo		
Parámetro	in / out	Descripción
window_name	in	Nombre de la ventana de trabajo
force	in	Dos valores disponibles: <ul style="list-style-type: none">• FALSE Valor por defecto. Si la Ventana esta abierta y existen tareas ejecutándose: No eliminara la ventana de trabajo.• TRUE Si la Ventana esta abierta y existen tareas ejecutándose: Deshabilitará las tareas y eliminara la ventana.
OPEN_WINDOW		
Abre la ventana de trabajo		
Parámetro	in / out	Descripción
window_name	in	Nombre de la ventana de trabajo
duration	in	Duración de la ventana de trabajo
force	in	Dos valores disponibles: <ul style="list-style-type: none">• FALSE Valor por defecto. Si la Ventana esta abierta y existen tareas ejecutándose: No eliminara la ventana de trabajo.• TRUE Si la Ventana esta abierta y existen tareas ejecutándose: Deshabilitará las tareas y eliminara la ventana.
CLOSE_WINDOW		
Cierra la ventana de trabajo		
Parámetro	in / out	Descripción
window_name	in	Nombre de la ventana de trabajo



Tabla 29: *pck_scheduler_windows_group*

PCK_SCHEDULER_WINDOWS_GROUP		
Paquete que contiene los procesos específicos a los grupos de ventana de trabajo.		
CREATE_WINDOW_GROUP		
Crea un grupo de ventanas de trabajo.		
Parámetro	in / out	Descripción
group_name	in	Nombre del grupo de ventanas de trabajo
window_list	in	Lista de ventanas asociadas al grupo.
comments	in	Descripción del grupo de ventanas de trabajo.
DROP_WINDOW_GROUP		
Crea una ventana de trabajo que esta asociada a una planificación específica.		
Parámetro	in / out	Descripción
group_name	in	Nombre del grupo de ventanas de trabajo
force	in	Valores disponibles: <ul style="list-style-type: none">• FALSE Valor por defecto. Si las Ventanas están abiertas y existen tareas ejecutándose: No eliminara el grupo de ventanas de trabajo.• TRUE Si las Ventanas están abiertas y existen tareas ejecutándose: Deshabilitará las tareas y eliminara el grupo de ventanas de trabajo.
ADD_WINDOW_GROUP_MEMBER		
Añade nuevas ventanas de trabajo al grupo de trabajo ya existente.		
Parámetro	in / out	Descripción
group_name	in	Nombre del grupo de ventanas de trabajo
window_list	in	Lista de ventanas que se asocian al grupo.
REMOVE_WINDOW_GROUP_MEMBER		
Quita las ventanas de trabajo del grupo de trabajo ya existente.		
Parámetro	in / out	Descripción
group_name	in	Nombre del grupo de ventanas de trabajo
window_list	in	Lista de ventanas que se desean quitar del grupo.

Scheduler Schedules

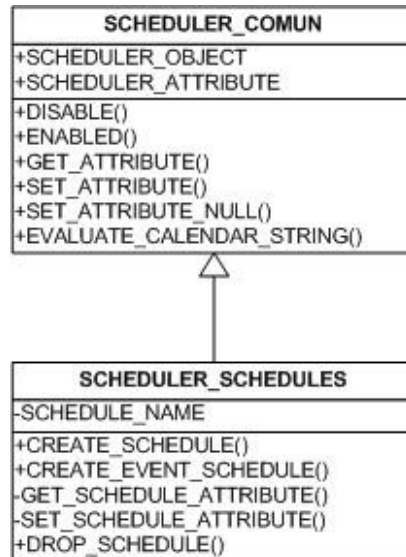


Ilustración 34: Scheduler Schedules

Se construye un paquete de base de datos que contiene los procesos específicos para el Objeto de Scheduler que esta asociado:

Tabla 30: *pck_scheduler_schedule*

PCK_SCHEDULER_SHEDULE		
Paquete que contiene los procesos específicos a las planificaciones y eventos.		
CREATE_SCHEDULE		
Crea una planificaciones.		
Parámetro	in / out	Descripción
schedule_name	in	Nombre de la planificación
start_date	in	Fecha a partir de la cual la planificación esta activa.
repeat_interval	in	Expresión del calendario.
end_date	in	Fecha a partir de la cual la planificación esta desactiva.
comments	in	Descripción de la planificación.
CREATE_EVENT_SCHEDULE		
Crea eventos.		
Parámetro	in / out	Descripción
schedule_name	in	Nombre del Evento
start_date	in	Fecha a partir de la cual el evento esta activo.
event_condition	in	Condiciones que hacen que se ejecute el evento.
queue_spec	in	Cola asociada al Evento.



end_date	in	Fecha a partir de la cual el evento esta desactivo.
comments	in	Descripción del Evento.
GET_SCHEDULE_ATTRIBUTE		
Obtiene el valor de un atributo de planificación		
Parámetro	in / out	Descripción
attribute	in	Nombre del atributo de schedule
value	out	Valor del atributo
SET_SCHEDULE_ATTRIBUTE		
Modifica el valor de un atributo de planificación		
Parámetro	in / out	Descripción
attribute	in	Nombre del atributo de schedule
value	out	Valor del atributo
DROP_SCHEDULE		
Elimina la planificación o evento		
Parámetro	in / out	Descripción
schedule_name	in	Nombre de la planificación / evento
force	in	Valores disponibles: <ul style="list-style-type: none"> FALSE Valor por defecto. Si existe algún objeto que referencia a la planificación o evento no se permitirá eliminar. TRUE Si existe algún objeto que referencia a la planificación o evento se deshabilitará y se eliminará la planificación / evento.

El Objeto Schedule tiene además la siguiente tabla de atributos que es específica y que no se puede modificar utilizando los procesos del paquete PCK_SCHEDULER_COMUN.

SCHEDULER ATTRIBUTE VALUES	
Nombre Atributo	Descripción
current_open_window	Parámetro de solo consulta. Devuelve la ventana de ejecución actualmente abierta
default_timezone	Zona Horaria usada por Scheduler para las expresiones.
event_expiry_time	Tiempo tras el cual un evento se expira.
log_history	Periodo de tiempo que Scheduler almacena los registros de ejecución.

Tabla 31: atributos de scheduler (scheduler attribute)

Scheduler Programs

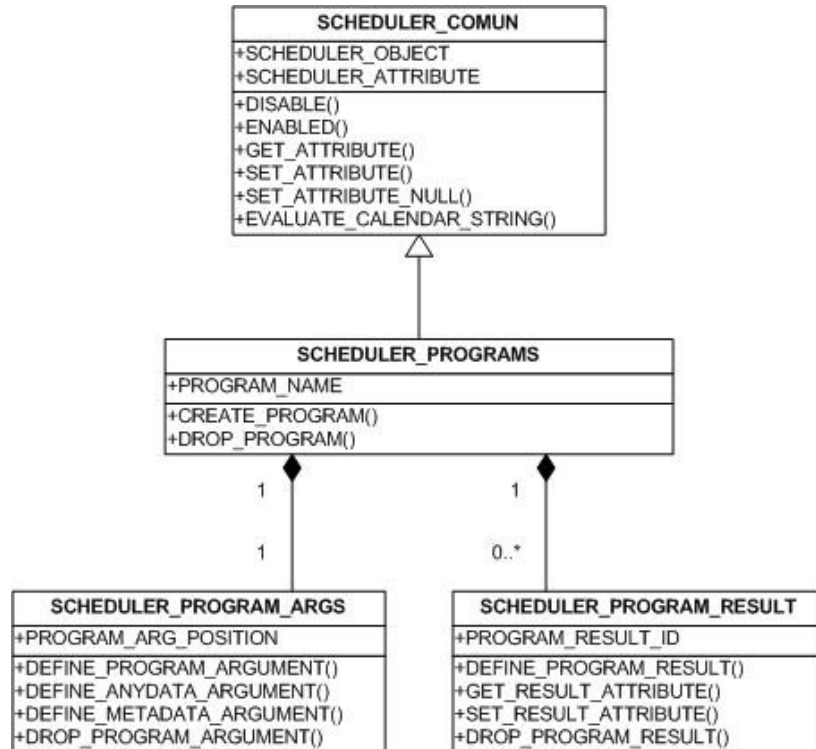


Ilustración 35: Scheduler Programs

Para este caso se podrían crear diferentes paquetes. Pero se incluirá todos los procesos en un mismo paquete ya que los Argumentos de Programa y Resultados están ligados a un programa.

Tabla 32: *pck_scheduler_program*

PCK_SCHEDULER_PROGRAM		
Paquete que contiene los procesos específicos para los programas.		
CREATE_PROGRAM		
Crea un programa		
Parámetro	in / out	Descripción
program_name	in	Nombre del programa
program_type	in	Tipo de programa: <ul style="list-style-type: none"> PLSQL_BLOCK STORED_PROCEDURE EXECUTABLE
program_action	in	Tipo de acción asociada al programa. La acción definida dependerá de tipo de programa que se especifique.
number_of_arguments	in	Número de argumentos del programa. Solo se indica si el tipo es <i>STORED-PROCEDURE</i> o <i>EXECUTABLE</i> .
enabled	in	Indica si el programa esta Activado (TRUE) o Desactivado (FALSE).



comments	in	Descripción del programa
DROP_PROGRAM		
Elimina el programa		
Parámetro	in / out	Descripción
program_name	in	Nombre del programa
force	in	Valores disponibles: <ul style="list-style-type: none">• FALSE Valor por defecto. Si existe algún objeto que referencia al programa no se permitirá eliminar.• TRUE Si existe algún objeto que referencia al programa se deshabilitará y se eliminará el programa.
DEFINE_PROGRAM_ARGUMENT		
Define los argumentos de los programas.		
Parámetro	in / out	Descripción
program_name	in	Nombre del programa
argument_position	in	Posición del argumento.
argument_name	in	Nombre del argumento.
argument_type	in	Tipo de valor del argumento (Valor reconocido por el lenguaje PL/SQL).
default_value	in	Valor por defecto del argumento.
DEFINE_ANYDATA_ARGUMENT		
Define el valor por defecto si se trata de un elemento complejo.		
Parámetro	in / out	Descripción
program_name	in	Nombre del programa
argument_position	in	Posición del argumento.
argument_name	in	Nombre del argumento.
argument_type	in	Tipo de valor del argumento.
default_value	in	Valor por defecto del argumento.
DEFINE_METADATA_ARGUMENT		
Se utiliza cuando el programa modifica atributos de objetos de Scheduler.		
Parámetro	in / out	Descripción
program_name	in	Nombre del programa
argument_position	in	Posición del argumento.
argument_name	in	Nombre del argumento.
metadata_attribute	in	Atributos de los Objetos Scheduler.
DEFINE_PROGRAM_RESULT		
Define los resultados de los programas		



Parámetro	in / out	Descripción
program_name	in	Nombre del programa
result_name	in	Nombre del fichero resultado tras la ejecución.
result_directory	in	Directorio donde se encuentra el fichero de resultado.
result_type	in	Tipo de Fichero.
result_delete	in	Indica si hay que eliminar el resultado. Tras su tratamiento.
DEFINE_PROGRAM_RESULT		
Define los resultados de los programas		
Parámetro	in / out	Descripción
program_name	in	Nombre del programa
result_function	in	Nombre de la función que genera el fichero de resultado.
result_directory	in	Directorio donde se encuentra el fichero de resultado.
result_type	in	Tipo de Fichero.
result_delete	in	Indica si hay que eliminar el resultado. Tras su tratamiento.
GET_RESULT_ATTRIBUTE		
Obtiene el valor de los atributos de resultado.		
Parámetro	in / out	Descripción
program_name	in	Nombre del programa
result_id	in	Identificador interno del resultado.
attribute	in	Nombre del atributo: <ul style="list-style-type: none">• result_name• result_funtion• result_directory• result_type• result_delete
value	out	Valor del atributo
SET_RESULT_ATTRIBUTE		
Modifica el valor de los atributos de resultado.		
Parámetro	in / out	Descripción
program_name	in	Nombre del programa
result_id	in	Identificador interno del resultado.
attribute	in	Nombre del atributo: <ul style="list-style-type: none">• result_name• result_funtion• result_directory• result_type• result_delete
value	in	Valor del atributo
DROP_PROGRAM_RESULT		



Elimina el resultado del programa.

Parámetro	in / out	Descripción
program_name	in	Nombre del programa
result_id	in	Identificador interno del resultado.

Scheduler Chains

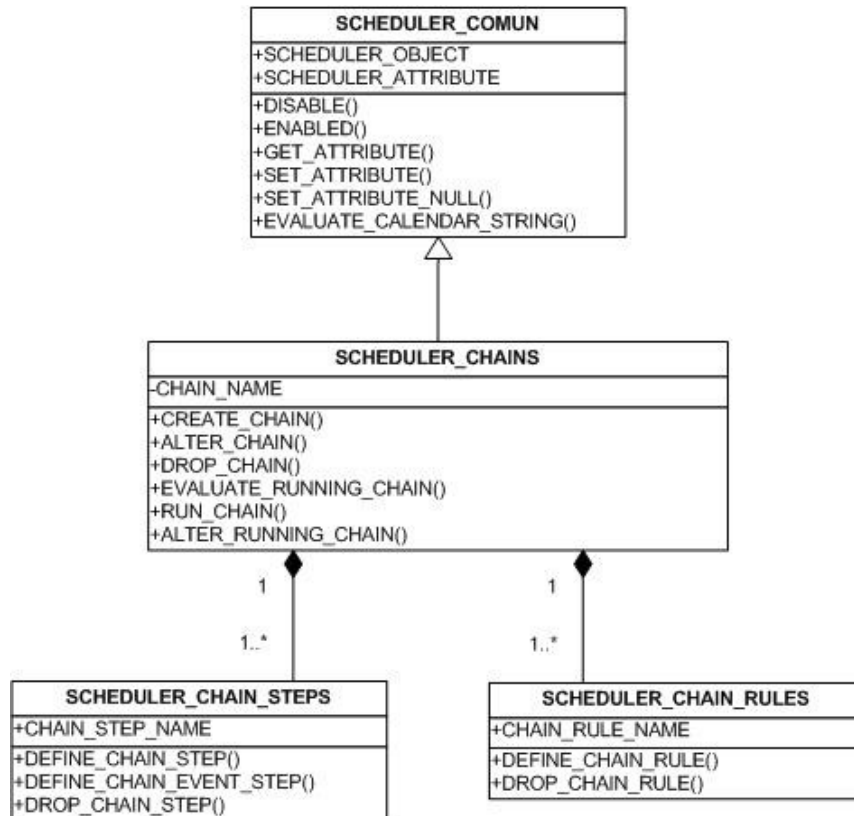


Ilustración 36: Scheduler Chains

Al igual que para los programas se pueden crear diferentes paquetes. Pero se incluirá todos los procesos en un mismo paquete ya que los Pasos y Condiciones de ejecución están ligados a un programa.

Tabla 33: *pck_scheduler_chains*

PCK_SCHEDULER_CHIANS		
Paquete que contiene los procesos específicos para los programas.		
CREATE_CHAIN		
Crea una cadena		
Parámetro	in / out	Descripción
chian_name	in	Nombre de la cadena
comments	in	Descripción de la cadena
DROP_CHAIN		
Elimina una cadena		
Parámetro	in / out	Descripción
chian_name	in	Nombre de la cadena



force	in	Valores disponibles: <ul style="list-style-type: none">• FALSE Valor por defecto. Si existe algún objeto que referencia a la cadena no se permitirá eliminar.• TRUE Si existe algún objeto que referencia a la cadena se deshabilitará y se eliminará la cadena.
-------	----	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ALTER_CHAIN

Modifica los atributos de los pasos asociados a la cadena.

Parámetro	in / out	Descripción
chain_name	in	Nombre de la cadena
step_name	in	Nombre del paso
attribute	in	Atributo a modificar <ul style="list-style-type: none">• PAUSE.• PAUSED_BEFORE• SKIP• RESTART_ON_FAILURE• RESTART_ON_RECOVERY
value	in	Valor que tomará el atributo

EVALUATE_RUNNING_CHAIN

Ejecuta una tarea que esta asociada a una cadena. Sirve para evaluar las condiciones de ejecución de la cadena.

Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea

RUN_CHAIN

Ejecuta la cadena de forma inmediata. Para ello Scheduler crea una Tarea automática de forma temporal.

Parámetro	in / out	Descripción
chain_name	in	Nombre de la cadena
start_steps	in	Lista de pasos a ejecutar cuando se ejecute la cadena
job_name	in	Nombre de la Tarea. Si no se especifica Scheduler asignara un nombre por defecto.

ALTER_RUNNING_CHAIN

Define los resultados de los programas

Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
step_name	in	Nombre del paso
attribute	in	Atributo a modificar <ul style="list-style-type: none">• PAUSE.• PAUSED_BEFORE• SKIP



		<ul style="list-style-type: none">• RESTART_ON_FAILURE• RESTART_ON_RECOVERY• STATE
value	in	Valor que tomará el atributo. Solo son validos los siguientes valores: TRUE, FALSE, NOT_STARTED, SUCCEEDED, FAILED. Para modificar el Atributo a FAILED hay que indicar el código de error.
DEFINE_CHAIN_STEP		
Crea un paso asociado a un Programa		
Parámetro	in / out	Descripción
chain_name	in	Nombre de la cadena
step_name	in	Nombre del paso
program_name	in	Programa asociado al paso
DEFINE_CHAIN_EVENT_STEP		
Crea un paso asociado a un Evento		
Parámetro	in / out	Descripción
chain_name	in	Nombre de la cadena
step_name	in	Nombre del paso
event_schedule_name	in	Nombre del Evento creado anteriormente en Scheduler
DROP_CHAIN_STEP		
Elimina el paso asociado a la cadena		
Parámetro	in / out	Descripción
chain_name	in	Nombre de la cadena
step_name	in	Nombre del paso
force	in	Valores disponibles: <ul style="list-style-type: none">• FALSE Valor por defecto. No permite borrar el paso si la cadena esta ejecutándose.• TRUE Si la cadena se esta ejecutando esperará hasta que finalice la ejecución. Si se ha ejecutado correctamente eliminara el paso.
DEFINE_CHAIN_RULE		
Crea una condición de Ejecución/finalización para la cadena		
Parámetro	in / out	Descripción
chain_name	in	Nombre de la cadena
condition	in	Condición de ejecución/finalización de la cadena.
action	in	Acciones que deben realizarse si se cumple la condición.
rule_name	in	Nombre de la condición.
comments	in	Descripción de la condición.



DROP_CHAIN_RULE		
Crea una condición de finalización para la cadena		
Parámetro	in / out	Descripción
chain_name	in	Nombre de la cadena
rule_name	in	Nombre condición
force	in	Valores disponibles: <ul style="list-style-type: none">• FALSE Valor por defecto. No permite borrar la condición si la cadena esta ejecutándose.• TRUE Si la cadena se esta ejecutando esperará hasta que finalice la ejecución. Si se ha ejecutado correctamente eliminara la condición.

Scheduler Jobs

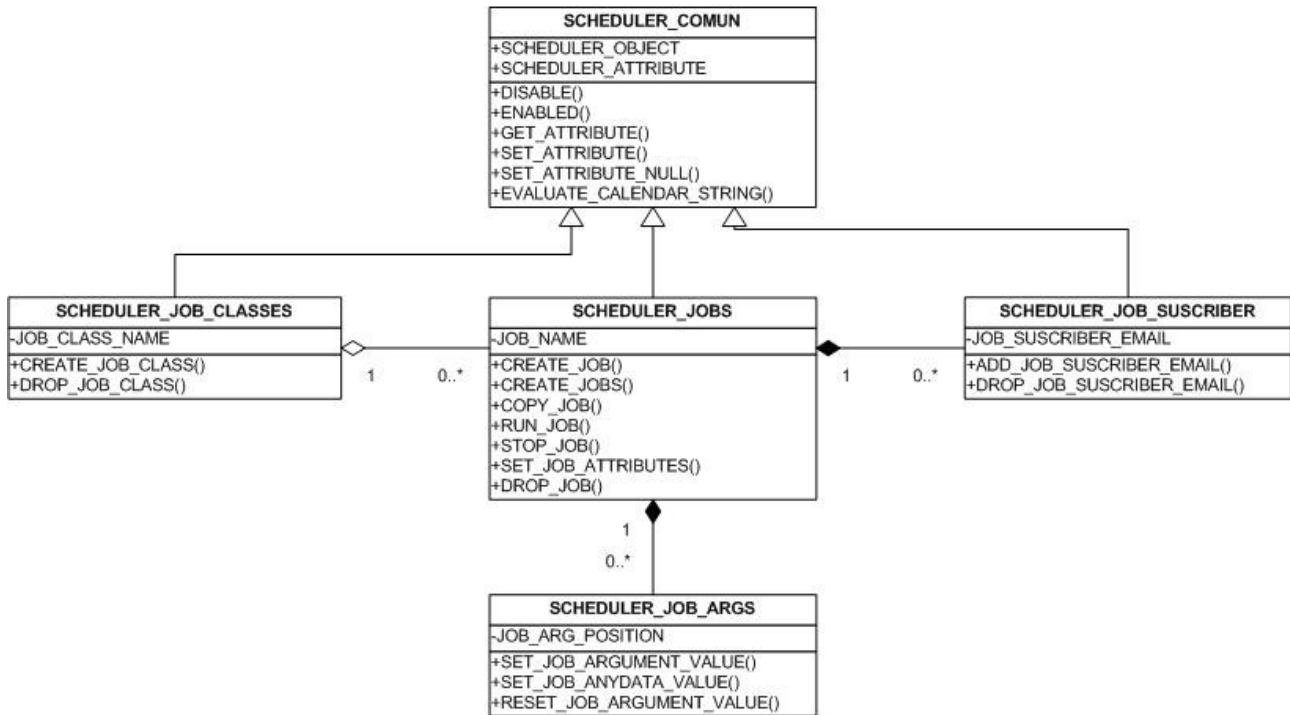


Ilustración 37: Scheduler Jobs

En este caso se crean dos paquetes:

- Clases de Tareas: Este paquete es independiente de la Tarea.
- Tareas: Dentro de este paquete se incluyen los procesos asociados a los Argumentos de las Tareas y a los Suscriptores de las Tareas, ya que ambos son dependientes de la Tarea.

Tabla 34: *pck_scheduler_job_classes*

PCK_SCHEDULER_JOB_CLASSES		
Paquete que contiene los procesos específicos para las Clases de Tareas		
CREATE_JOB_CLASS		
Crea una clase de Tarea		
Parámetro	in / out	Descripción
job_class_name	in	Nombre de la clase de tarea
resource_consumer_group	in	Grupo de consumo que esta asociado la clase de tarea. Excluyente del Servicio
service	in	Servicio asociado a la clase de tarea. Excluyente del Grupo de Recursos
logging_level	in	Nivel de registro
log_history	in	Número de días que se guardaran los registros
comments	in	Descripción de la clase de tarea



DROP_JOB_CLASS		
Elimina una cadena		
Parámetro	in / out	Descripción
job_class_name	in	Nombre de la clase de tarea
force	in	Valores disponibles: <ul style="list-style-type: none"> FALSE Valor por defecto. Si existe algún objeto que referencia a la clase de tarea no se permitirá eliminar. TRUE Si existe algún objeto que referencia a la clase de tarea se le asociara la clase por defecto y se eliminará la clase de tarea.

Tabla 35: *pck_scheduler_jobs*

PCK_SCHEDULER_JOBS		
Paquete que contiene los procesos específicos para las Tareas		
CREATE_JOB		
Crea una tarea simple sin hacer uso de cualquier otro objeto de Scheduler		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
job_type	in	Tipo de acción: PLSQL_BLOCK, STORED_PROCEDURE, EXECUTABLE o CHAIN
job_action	in	Acción a ejecutar. Depende del Tipo seleccionada.
number_of_arguments	in	Número de argumentos.
start_date	in	Fecha a partir de la cual la tarea esta activa.
end_date	in	Fecha a partir de la cual la tarea esta desactiva.
repeat_interval	in	Expresión del calendario.
job_class	in	Clase de Tarea asociada. Por defecto se asocia: DEFAULT_JOB_CLASS
enabled	in	Indica si la tarea esta activa o desactiva
comments	in	Descripción de la tarea.
CREATE_JOB		
Crea una tarea utilizando un programa y una planificación		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
program_name	in	Nombre del Programa
schedule_name	in	Nombre de la planificación.



job_class	in	Clase de Tarea asociada. Por defecto se asocia: DEFAULT_JOB_CLASS
enabled	in	Indica si la tarea esta activa o desactiva
comments	in	Descripción de la tarea.
CREATE_JOB		
Crea una tarea utilizando un programa y sin planificación		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
program_name	in	Nombre del Programa
start_date	in	Fecha a partir de la cual la tarea esta activa.
end_date	in	Fecha a partir de la cual la tarea esta desactiva.
repeat_interval	in	Expresión del calendario.
job_class	in	Clase de Tarea asociada. Por defecto se asocia: DEFAULT_JOB_CLASS
enabled	in	Indica si la tarea esta activa o desactiva
comments	in	Descripción de la tarea.
CREATE_JOB		
Crea una tarea utilizando una planificación.		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
job_type	in	Tipo de acción: PLSQL_BLOCK, STORED_PROCEDURE, EXECUTABLE o CHAIN.
job_action	in	Acción a ejecutar. Depende del Tipo seleccionada.
number_of_arguments	in	Número de argumentos.
schedule_name	in	Nombre de la planificación.
job_class	in	Clase de Tarea asociada. Por defecto se asocia: DEFAULT_JOB_CLASS
enabled	in	Indica si la tarea esta activa o desactiva
comments	in	Descripción de la tarea.
CREATE_JOB		
Crea una tarea asociado a un evento		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
job_type	in	Tipo de acción: PLSQL_BLOCK, STORED_PROCEDURE, EXECUTABLE o CHAIN
job_action	in	Acción a ejecutar. Depende del Tipo seleccionada.
number_of_arguments	in	Número de argumentos.



start_date	in	Fecha a partir de la cual la tarea esta activa.
end_date	in	Fecha a partir de la cual la tarea esta desactiva.
event_condition	in	Condiciones que hacen que se ejecute el evento.
queue_spec	in	Cola asociada al Evento.
job_class	in	Clase de Tarea asociada. Por defecto se asocia: DEFAULT_JOB_CLASS
enabled	in	Indica si la tarea esta activa o desactiva
comments	in	Descripción de la tarea.
CREATE_JOB		
Crea una tarea asociado a un evento y un Programa		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
program_name	in	Nombre del Programa
start_date	in	Fecha a partir de la cual la tarea esta activa.
end_date	in	Fecha a partir de la cual la tarea esta desactiva.
event_condition	in	Condiciones que hacen que se ejecute el evento.
queue_spec	in	Cola asociada al Evento.
job_class	in	Clase de Tarea asociada. Por defecto se asocia: DEFAULT_JOB_CLASS
enabled	in	Indica si la tarea esta activa o desactiva
comments	in	Descripción de la tarea.
CREATE_JOBS		
Crea múltiples tareas a partir de una cadena que define los atributo de las tareas.		
Parámetro	in / out	Descripción
jobdef_array	in	Lista con las definiciones de las tareas.
commit_semantics	in	Indica cuando realizar el commit: <ul style="list-style-type: none">• STOP_ON_FIRST_ERROR• TRANSACTIONAL• ABSORB_ERROR
COPY_JOB		
Copia una tarea		
Parámetro	in / out	Descripción
old_job	in	Nombre de la Tarea a copiar
new_job	in	Nombre de la nueva Tarea
RUN_JOB		
Ejecuta una tarea		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea



use_current_session	in	Indica si ha de usar la sesión actual o no. Su valor por defecto es TRUE.
STOP_JOB		
Detiene la ejecución de una tarea.		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
force	in	Valores disponibles: <ul style="list-style-type: none">FALSE Valor por defecto. Detiene la tarea y acciones asociadas.TRUE Detiene la tarea pero no las acciones asociadas a la tarea.
commit_semantics	in	Indica cuando realizar el commit: <ul style="list-style-type: none">STOP_ON_FIRST_ERRORTRANSACTIONALABSORB_ERROR
SET_JOB_ATTRIBUTES		
Crea múltiples tareas a partir de una cadena que define los atributo de las tareas.		
Parámetro	in / out	Descripción
jobdef_array	in	Lista de atributos a modificar.
commit_semantics	in	Indica cuando realizar el commit: <ul style="list-style-type: none">STOP_ON_FIRST_ERRORTRANSACTIONALABSORB_ERROR
DROP_JOB		
Elimina la tarea		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
force	in	Valores disponibles: <ul style="list-style-type: none">FALSE Valor por defecto. Elimina la tarea.TRUE Elimina la tarea si no se esta ejecutando.
defer	in	Valores disponibles: <ul style="list-style-type: none">FALSE (Valor por defecto.) Elimina la tarea.TRUE Espera a que finalice todas las tareas que están en ejecución.
commit_semantics	in	Indica cuando realizar el commit:



		<ul style="list-style-type: none">• STOP_ON_FIRST_ERROR (Valor por defecto)• TRANSACTIONAL• ABSORB_ERROR
SET_JOB_ARGUMENT_VALUE		
Cambia el valor de los argumentos		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
argument_position	in	Posición del argumento.
argument_value	in	Valor del argumento.
SET_JOB_ANYDATA_VALUE		
Cambia el valor de los argumentos complejos		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
argument_position	in	Posición del argumento.
argument_value	in	Valor del argumento.
RESET_JOB_ARGUMENT_VALUE		
Limpia el valor del argumento.		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
argument_position	in	Posición del argumento.
ADD_JOB_SUSCRIBER_EMAIL		
Añade una suscripción a la Tarea		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
suscriber_email	in	Correo donde se deben enviar las notificaciones
condition	in	Condición de envío de notificación. Depende del estado en que quede la tarea tras su ejecución: COMPLETED, SUCCEDED, FAILED, STOPPED y STALLED
error	in	Si la condición seleccionada es STALLED. Hay que indicar el Código de Error.
DROP_JOB_SUSCRIBER_EMAIL		
Elimina la suscripción de la Tarea		
Parámetro	in / out	Descripción
job_name	in	Nombre de la tarea
suscriber_email	in	Correo donde se deben enviar las notificaciones

5.2 Implementación Vista – Controlador

Es necesario realizar una introducción a la hermanita de desarrollo que se va a utilizar, antes de entrar en detalle a la Implementación de la interfaz gráfica y de los Controladores correspondientes para el funcionamiento de los formularios.

Introducción Oracle Forms Developer

Oracle Forms Developer proporcionan un marco de aplicación completo para desplegar las aplicaciones en Internet así como la identificación en redes corporativas. Ofrece un entorno de desarrollo rápido de aplicaciones (RAD – *Rapid Application Development*).

Las aplicaciones creadas con *Oracle Forms* se escriben y se distribuyen en Java. Permite a desarrolladores centrarse exclusivamente en la implementación de la lógica de negocio sin la necesidad de escribir ningún código Java.

El entorno de desarrollo RAD ofrece un conjunto de herramientas que incluye muchos asistentes y utilidades que permite desarrollar aplicaciones a gran velocidad. Algunas de las características que posee el entorno de desarrollo son:

- **Bloque de Datos** : Un *Bloque de Datos* corresponde a una tabla de base de datos. Para cada tabla que se muestre en el formulario se creará un nuevo Bloque de Datos.

Hay numerosas formas de crear un Bloque de Datos.

Una forma es definir manualmente la tabla base y las columnas. Aunque esto puede ser posible, puede ser muy tedioso.

La otra es utilizar el asistente que ofrece Oracle Forms para crear *Bloque de Datos*, dicho asistente guía al desarrollador a través de sus pasos, a la elección de una tabla de base de datos y las columnas que se quiera introducir. Una vez creado el Bloque de Datos, se puede volver a entrar al asistente para su modificación.

- **Editor Diseño** : El *Editor de Diseño* se utiliza para posicionar los campos del Bloque de Datos sobre un Lienzo.

Al igual que en los bloques de datos, la distribución de los campos sobre el lienzo se puede realizar de forma manual o utilizando el asistente que Oracle Forms proporciona.

- **Paleta de propiedades** : La *Paleta de Propiedades* permite establecer las propiedades de los objetos que se crean. Cuando se selecciona un objeto, en el Editor de Diseño o en el Navegador de Objetos, la *Paleta de propiedades* se actualiza mostrando las propiedades del objeto seleccionado.

Se puede invocar tantas Paletas de Propiedades como sean necesarias para comparar propiedades de diferentes objetos.

- **Editor PL/SQL** : El *Editor PL/SQL* permite escribir código PL/SQL desde dentro de los objetos. La interfaz de edición permite la compilación y depuración del código introducido.

- **Biblioteca de Objetos** : Las *Bibliotecas de Objetos* permite la re-utilización de objetos y módulos de programación. De esta forma se puede hacer el uso de estándares para el desarrollo de la aplicación.

- **Paquetes Integrados** : Se proporciona varios *Paquetes Integrados* que contiene código PL/SQL ya creado que el desarrollador puede utilizar.

Se proporciona una gran variedad de widgets nativos y controles por defecto para un rápido desarrollo de interfaces de usuario convincentes y con codificación mínima. Algunos de los widgets que se pueden encontrar son: Árboles Jerárquicos, Lienzos con pestañas, Casillas de control, Listas desplegables,

campos calculados, campos resumen, etc...

Oracle Forms permite extender y personalizar su entorno de desarrollo utilizando componentes java.

Las aplicaciones creadas en Oracle Forms se ejecutan en un Applet de Java que proporciona la interfaz necesaria para la ejecución de formularios, interacción con el usuario y retroalimentación visual. El mismo Applet de Java se utiliza para todos los formularios, por lo tanto, se descarga una sola vez y se almacena en la caché del cliente y así esta disponible para su posterior ejecución.

Para ejecutar el Applet de Java en un navegador, es necesario contar con una máquina virtual Java (JVM) instalado. La JVM esta instalada en el cliente y es independiente de la plataforma.

La ejecución de un formulario mantiene una conexión abierta a la base de datos con el nombre del Cliente. Se crea una sesión cuando el usuario accede a la aplicación. Y se cierra cuando el usuario cierra la aplicación.











Barra de herramientas

Anteriormente se ha mencionado que los formularios se ejecutan en un Applet de Java que permite la interacción con el usuario. Si no se especifica lo contrario el Applet mostrará una barra de hermanitas por defecto que permitirá realizar las acciones comunes de los formularios.



Ilustración 38: Barra de Herramientas

Dicha barra de herramientas se puede personalizar. Los botones que se quiere destacar son:

-  **Guardar** : Almacena los nuevos registros, las modificaciones y las eliminaciones. Es como realizar un *commit* en la base de datos.
-  **Salir** : Se sale de la ejecución del Formulario.
-  **Ejecutar Consulta** : Ejecuta una consulta sobre la base de datos en función del contenido del formulario, si no existe ningún contenido, permite consultar todos los registros.
-  **Introducir Consulta** : Permite introducir una consulta, para obtener los registros de base de datos que cumplan una determinada condición.
-  **Bloque Anterior** : Navega al anterior Bloque de Datos.
-  **Registro Anterior** : Navega al anterior registro de un Bloque de Datos.
-  **Registro Siguiente** : Navega al siguiente registro de un Bloque de Datos.
-  **Bloque Siguiente** : Navega al siguiente Bloque de Datos.
-  **Insertar Registro** : Incluye un nuevo registro en el Bloque de Datos.
-  **Suprimir Registro** : Suprime un registro del Bloque de Datos.

Mantenimiento Clase de Tarea

Si se echa un vistazo al diseño que se ha definido para este Formulario:

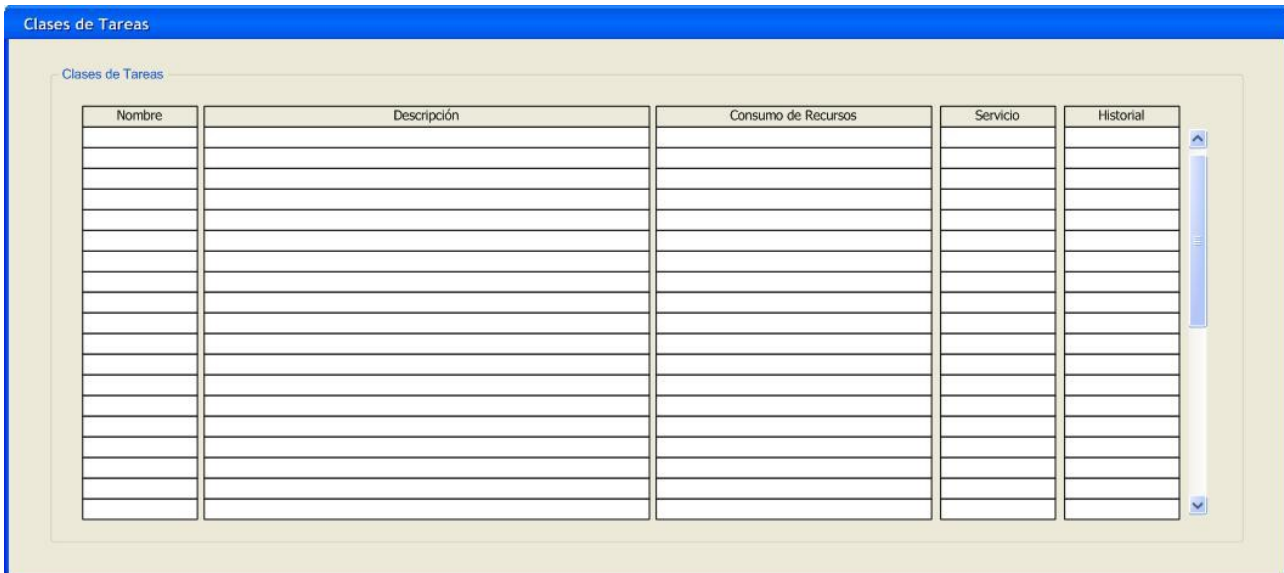


Ilustración 39: Mantenimiento Clase de Tarea

Se crea un solo **Bloque de Datos** sobre la vista `dba_scheduler_job_classes`. Como se trata de una vista del Sistema, Oracle no permite realizar Inserciones, Modificaciones ni Supresión de registros. Para ello se debe hacer uso del paquete `dbms_scheduler`.

Por motivos de Seguridad de Oracle no permite ejecutar dichos paquetes desde los Formularios, por ese motivo se han implementado los paquetes de base de datos.

Se deberá incluye tres controles sobre el Bloque de Datos para que llamen a los procesos de base de datos correspondientes y seguidamente detener la ejecución para que no salga el error de Oracle de que no se permite insertar, modificar o suprimir el registro.

Los controles a incluir son:

- **pre-insert:** : Este control permite ejecutar cualquier código PL/SQL antes de realizar la inserción sobre la tabla/vista asociada al Bloque de Datos donde se ha incluido este controlador.

En este caso deberá llamar al procedimiento `create_job_class` del paquete `pck_scheduler_job_class`.

Si el proceso de creación encuentra algún error, dicho error se muestra por pantalla haciendo uso del procedimiento ALERT que ofrece Oracle.

Tanto si el proceso termina con error como si termina satisfactoriamente se ejecuta el procedimiento `FORM_TRIGGER_FAILURE` que detiene cualquier ejecución.

- **Pre-update** : Este control permite ejecutar cualquier código PL/SQL antes de realizar la modificación sobre la tabla/vista asociada al Bloque de Datos donde se ha incluido este controlador.

En este caso deberá llamar al procedimiento `set_attribute` del paquete `pck_scheduler_comun`.

Si el proceso de modificación encuentra algún error, dicho error se muestra por pantalla haciendo uso del procedimiento ALERT que ofrece Oracle.

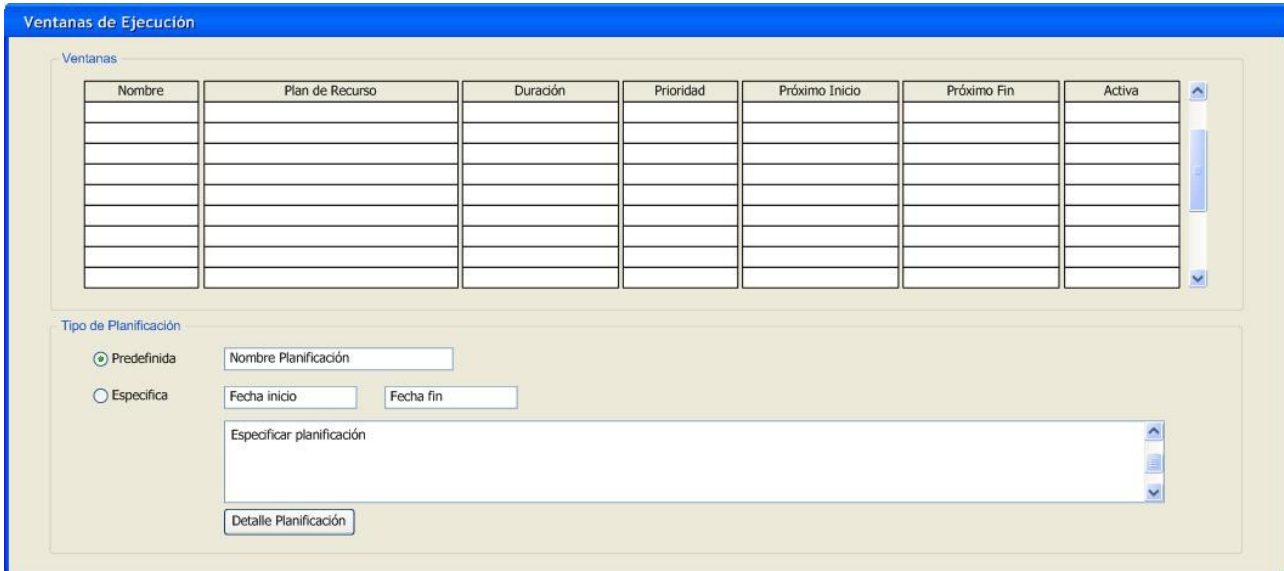
Tanto si el proceso termina con error como si termina satisfactoriamente se ejecuta el procedimiento `FORM_TRIGGER_FAILURE` que detiene cualquier ejecución.



- **Pre-delete** : Este control permite ejecutar cualquier código PL/SQL antes de realizar la supresión sobre la tabla/vista asociada al Bloque de Datos donde se ha incluido este controlador.
En este caso deberá llamar al procedimiento *drop_job_class* del paquete *pck_scheduler_job_class*.
Si el proceso de supresión encuentra algún error, dicho error se muestra por pantalla haciendo uso del procedimiento ALERT que ofrece Oracle.
Tanto si el proceso termina con error como si termina satisfactoriamente se ejecuta el procedimiento FORM_TRIGGER_FAILURE que detiene cualquier ejecución.

Mantenimiento Ventanas de Trabajo

Si se echa un vistazo al diseño que se ha definido para este Formulario:



Nombre	Plan de Recurso	Duración	Prioridad	Próximo Inicio	Próximo Fin	Activa

Tipo de Planificación

Predefinida

Especifica

Ilustración 40: Mantenimiento Ventanas de Trabajo

Siguiendo el mismo razonamiento que anterior formulario se crea un **Bloque de Datos** sobre la vista *dba_scheduler_windows*.

En este caso los campos referidos al Tipo de Planificación se incluirán en un Marco a parte, para que solo se visualice un registro y no múltiples registros se cambia la propiedad correspondiente.

Se incluye en dicho marco un grupo de botones de control, que habilita los campos correspondientes según la planificación que se escoja.

El botón “Detalle Planificación” implementa la llamada al formulario de detalle de planificaciones, pasando por parámetro los campos correspondientes.

Al igual que el Mantenimiento anterior se incluyen tres controladores sobre el Bloque de Datos:

- **pre-insert** : llama al proceso *pck_scheduler_windows.create_window*.
- **Pre-update** : llama al proceso *pck_scheduler_comun.set_attribute*
- **pre-delete** : llama al proceso *pck_scheduler_windows.drop_window*

Mantenimiento de Planificaciones

El formulario diseñado es el siguiente:

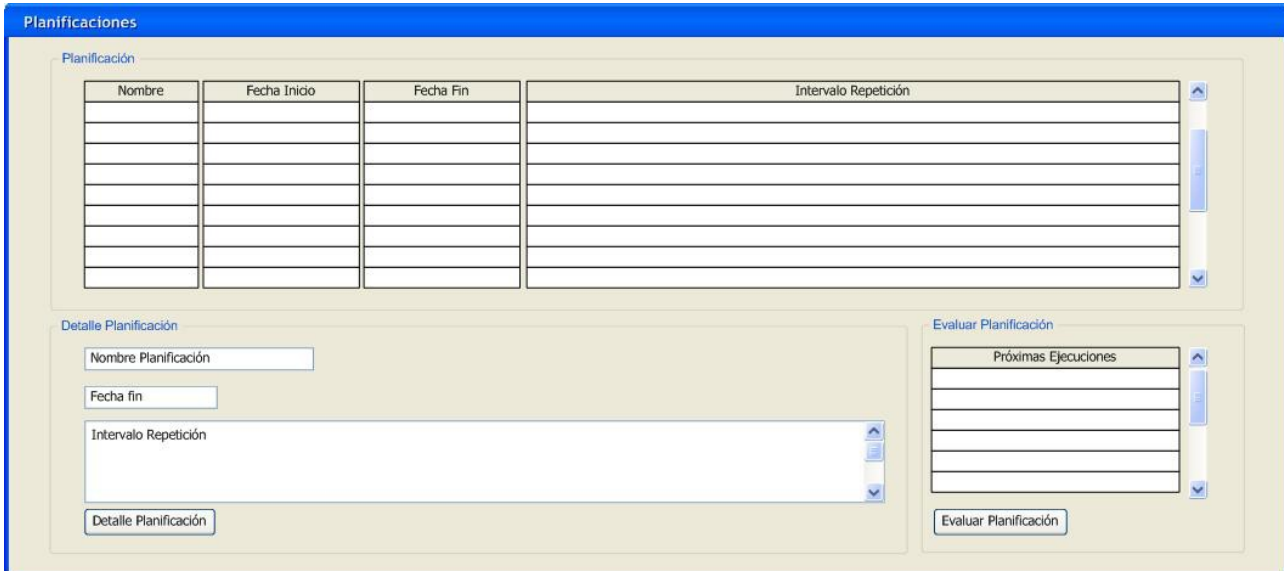


Ilustración 41: Mantenimiento Planificaciones

Se crea un **Bloque de datos** sobre la vista `dba_scheduler_schedules`. Se crean en el mismo bloque una copia de los campos modificando en los campos copiados el número de registros visualizados y indicando que son elementos de referencia de los campos anteriores. De esta forma se consigue que al modificar un campo tanto de la tabla como del marco detalle afecte a los dos campos a la vez.

Se crea un Bloque de Datos que no va sobre ninguna tabla de base de datos donde se incluirá un campo que muestra las próximas ejecuciones.

Los botones que se incluyen son:

- Botón “Detalle Planificación” : llama al formulario correspondiente.
- Botón “Evaluar Planificación” : este botón implementa un código PL/SQL que llama dentro de un bucle al procedimiento `scheduler_comun.evaluate_calendar_string` el valor retornado lo ira relleno en el campo del Bloque de Datos “Evaluar Planificación”

En el Bloque de Datos correspondiente a la “Planificación” se incluyen los siguientes controles:

- **pre-insert** : llama al proceso `pck_scheduler_schedules.create_schedule`
- **Pre-update** : llama al proceso `pck_scheduler_schedules.set_schedule_attribute` o `pck_scheduler_comun.set_attribute`
- **pre-delete** : llama al proceso `pck_scheduler_schedules.drop_schedule`

Detalle Planificación

El Formulario diseñado es:

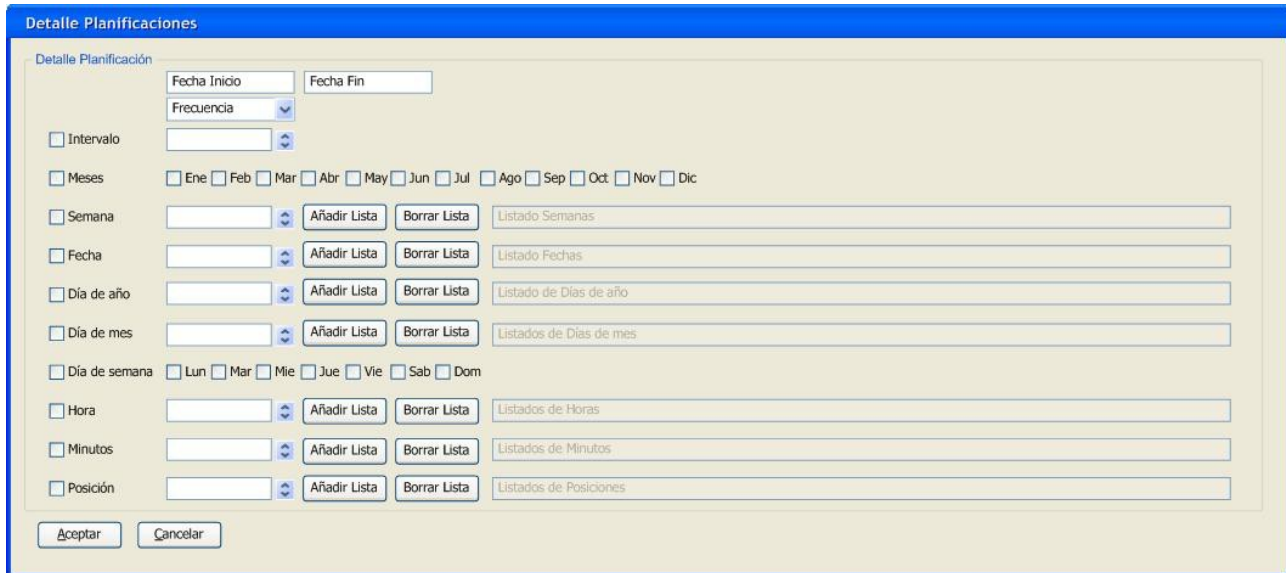


Ilustración 42: Detalle Planificación

A diferencia de los otros formularios este no va sobre ninguna tabla de base de datos. Este formulario se abriría como ventana emergente y es el único también que no heredaría la barra de herramientas.

Se crea un Bloque de Datos, se incluye en el todos los elementos del formulario. Los controles que tiene el formulario son:

- **when-new-form-instance** : Controlador que ejecuta código PL/SQL antes de que el formulario se abra por completo. En este disparador se lee los parámetros recibidos y rellena los campos correspondientes.

Los botones que se implementan son:

- **Añadir Lista** : Incluye el valor del campo al la lista de valores de dicho campo.
- **Borrar lista** : Borra la lista creada para el campo
- **Aceptar** : Rellena los parámetros de entrada con los datos introducidos en los campos y sale del formulario.
- **Cancelar** : Cancela los cambios y sale del formulario.

La comunicación entre este formulario y los demás se realiza mediante variables Globales. Estas variables son campos creados a nivel de aplicación, es decir que una vez creados pueden ser consultados por cualquier formulario.

Para evitar una sobrecarga de memoria, se crean estas globales antes de realizar la llamada al formulario de Detalle. Y se borrarán una vez se cierra el formulario de Detalle y se hayan tratado el valor de los globales.

Mantenimiento de Eventos

El Formulario diseñado es:

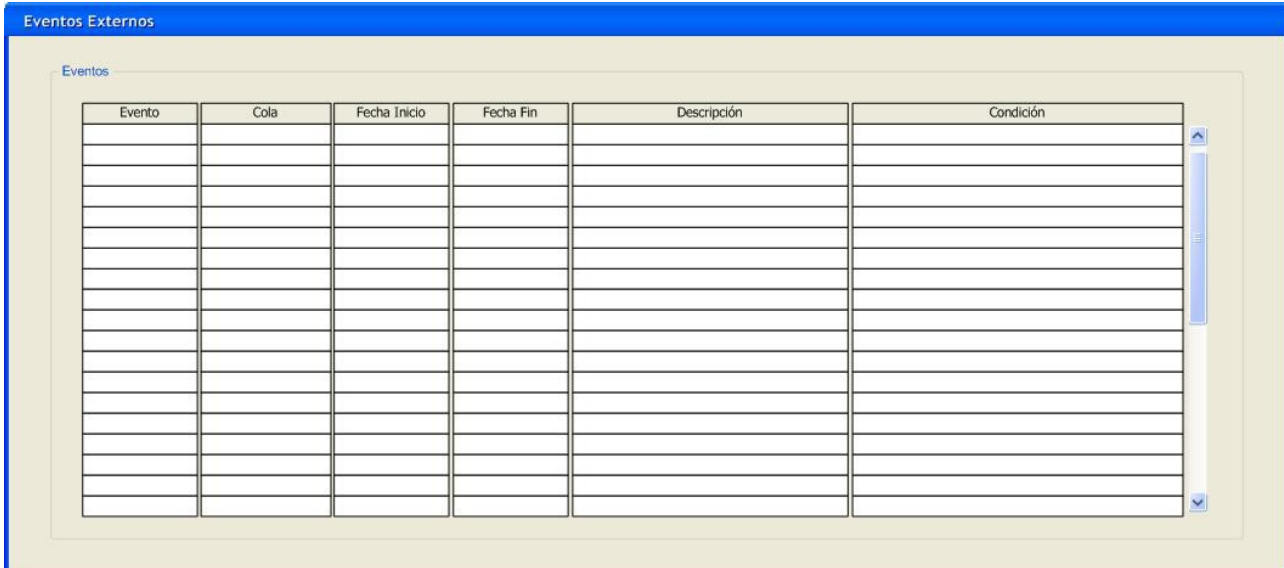


Ilustración 43: Mantenimiento de Eventos

Se crea un **Bloque de datos** sobre la vista `dba_scheduler_schedules`. Se incluye en dicho Bloque de Datos los siguientes controles:

- **pre-insert** : llama al proceso `pck_scheduler_schedules.create_event_schedule`
- **Pre-update** : llama al proceso `pck_scheduler_schedules.set_schedule_attribute` o `pck_scheduler_comun.set_attribute`
- **pre-delete** : llama al proceso `pck_scheduler_schedules.drop_schedule`

Gestor de Programas

Este formulario esta formado por un lienzo con varias pestañas. Además se incluye un componente complejo como es el Árbol Jerárquico. Antes de entrar al diseño de cada una de las pestañas se realizará una introducción a los Árboles Jerárquicos.

En ocasiones se ve la necesidad de recuperar los datos de una tabla o vista con relación recursiva, para mostrar pertenencias o jerarquías. Tales jerarquías se representan normalmente a través de estructuras en árbol, en donde la raíz (o nodo principal) tiene ramas (nodos descendientes) y cada una de estas ramas puede a su vez generar más ramas, hasta llegar aun punto final en donde no hay más divisiones. Los nodos finales son llamados hojas.

El lenguaje SQL de Oracle permite la posibilidad de recuperar los datos de una tabla, vista, o tabla externa, simulando el recorrido de un árbol, el cual puede hacerse comenzando por la raíz hasta llegar a todas las hojas, o desde las hojas hasta llegar a la raíz.

Para ello la estructura SQL básica es:

```
SELECT      [columna],                                LEVEL
FROM        [tabla]
START WITH  [columna]='valor'                          nodo      raíz'
CONNECT BY PRIOR [columna 1] = [columna 2];
```

Donde:

- LEVEL : Indica el nivel donde se encuentra el registro, tras la construcción del árbol.
- START WIDTH : Indica el registro raíz, es decir, donde empieza el árbol.
- CONNECT BY PRIOR : indica las columnas en las cuales existe la relación padre-hijo.

Ahora que ya se sabe como se construye una consulta jerárquica en Oracle, se procede a indicar como se construye este Gestor de Programa. Los diseños que se tiene son:

Ilustración 44: Gestor Programas - Datos Generales

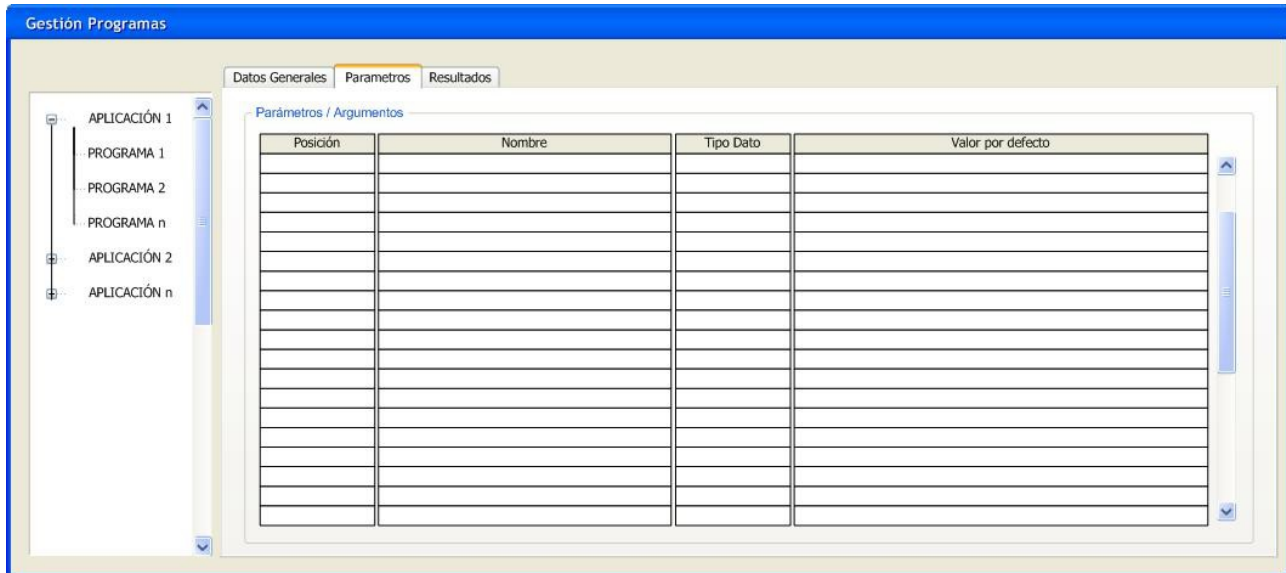


Ilustración 45: Gestor Programas - Parámetros

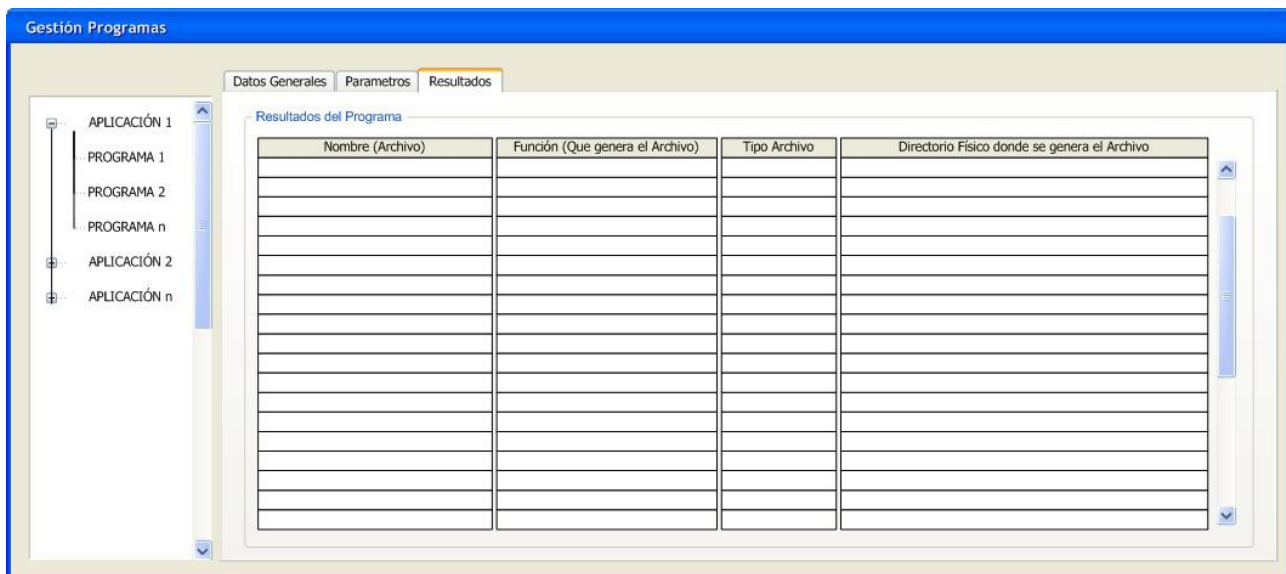


Ilustración 46: Gestor Programas - Resultados

Se crea un **Bloque de Datos** que contendrá el Árbol Jerárquico. Dicho árbol se creará mediante una consulta jerárquica sobre la tabla *dba_scheduler_program*. Las hojas de este árbol son los programas que se han creado. Al seleccionar una hoja se consultarán todos sus datos.

En la Pestaña “*Datos Generales*”, contendrá un **Bloque de Datos** que estará asociada también a la tabla *dba_scheduler_program*, a diferencia de los anteriores formularios este solo muestra un registro.

Se incluye en el bloque los siguientes controles:

- **pre-insert** : llama al proceso *pck_scheduler_program.create_program*
- **Pre-update** : llama al proceso *pck_scheduler_comun.set_attribute*.



- **pre-delete** : llama al proceso *pck_scheduler_program.drop_program*.

El botón “Crear Cadena Simple” implementa el código PL/SQL necesario para generar una Cadena con un solo Paso que llama a el programa actual y las Condiciones necesarias para poder ejecutar la cadena.

En la Pestaña “Parámetros”, contendrá un **Bloque de Datos** que esta asociada a la tabla *dba_scheduler_program_args*, Se incluye en el bloque los siguientes controles:

- **pre-insert** : llama al proceso *pck_scheduler_program_args.define_program_argument*
- **Pre-update** : llama al proceso *pck_scheduler_program_args.define_anydata_argument*
- **pre-delete** : llama al proceso *pck_scheduler_program_args.drop_program_argument*

En la Pestaña “Resultados”, contendrá un **Bloque de Datos** que esta asociada a la tabla *dba_scheduler_program_result*, Se incluye en el bloque los siguientes controles:

- **pre-insert** : llama al proceso *pck_scheduler_program_result.define_program_result*
- **Pre-update** : llama al proceso *pck_scheduler_program_result.set_result_attribute*
- **pre-delete** : llama al proceso *pck_scheduler_program_result.drop_program_result*

Gestor de Cadenas

Como en el Gestor de Programas, este Gestor contiene un elemento complejo como lo es el árbol jerárquico. Los diseños que se tiene para este Gestor son:

Nombre Cadena	Descripción Cadena	Estado	Aplicación	Responsable

Nombre Paso	Tipo Paso	Acción	Ejecutable (Campo Recuperado)

Nombre Paso	Acción

Nombre Paso	Acción

Ilustración 47: Gestor Cadenas - Gestión

Acción Finalización	Condición Finalización Cadena

Ilustración 48: Gestor Cadenas - Datos Generales

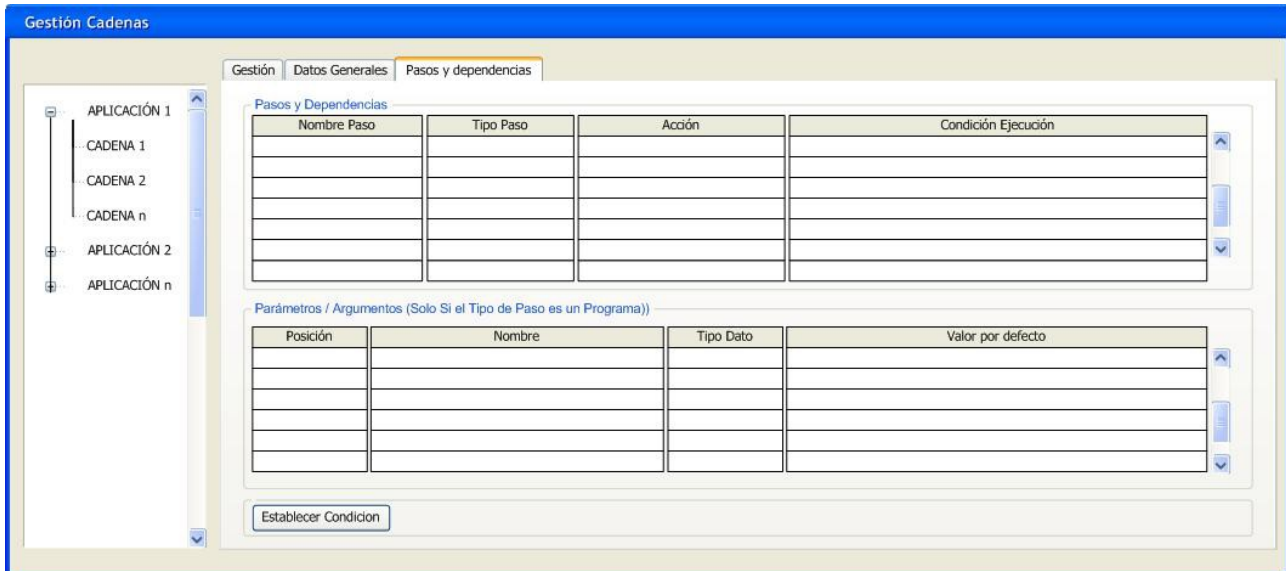


Ilustración 49: Gestor Cadenas - Pasos y Dependencias

Se crea un **Bloque de Datos** que contendrá el Árbol Jerárquico. Dicho árbol se creará mediante una consulta jerárquica sobre la tabla *dba_scheduler_chains*. Las hojas de este árbol son las cadenas que se han creado. Al seleccionar una hoja se consultarán todos sus datos.

En la Pestaña “Gestión”, contiene varios **Bloque de Datos** que solo son de visualización, es decir no se permite ninguna acción sobre el bloque:

- Cadena : Este Bloque va sobre la vista *dba_scheduler_chains*.
- Pasos : Este Bloque va sobre la vista *dba_scheduler_chain_steps* y se trata de un bloque detalle del anterior.
- Paso Predecesor : Este Bloque va sobre la vista *dba_scheduler_chain_steps* y se trata de un bloque detalle del de Pasos. Se modifica la consulta para que recupere el paso Anterior si existe.
- Paso Sucesor : Este Bloque va sobre la vista *dba_scheduler_chain_steps* y se trata de un bloque detalle del de Pasos. Se modifica la consulta para que recupere el paso Anterior si existe.

En dicha pestaña se incluye tres botones:

- Activar Cadena : llama a *pck_scheduler_comun.enabled*.
- Desactivar Cadena : llama a *pck_scheduler_comun.disabled*.
- Copiar Cadena : contiene el código PL/SQL necesario para copiar una cadena.

En la Pestaña “Datos Generales”, contiene dos **Bloque de Datos**:

- Datos generales : Este Bloque va sobre la vista *dba_scheduler_chains*. Se incluye los controles:
 - **pre-insert** : llama al proceso *pck_scheduler_chains.create_chain*
 - **Pre-update** : llama al proceso *pck_scheduler_comun.set_attribute*

- **pre-delete** : llama al proceso `pck_scheduler_chans.drop_chian`
- Condición Ejecución : Este Bloque va sobre la vista `dba_scheduler_chain_rules`. Se trata de un bloque detalle del anterior. Se incluye los siguientes controles:
 - **pre-insert** : llama al proceso `pck_scheduler_chain_rules.define_chain_rule`
 - **pre-delete** : llama al proceso `pck_scheduler_chain_rules.drop_chain_rule`

El botón Establecer condición llamará al formulario “Establecer Condición”.

En la Pestaña “Pasos y Dependencias”, contiene dos **Bloque de Datos**:

- Pasos y Dependencias : Este Bloque va sobre la vista `dba_scheduler_steps`. Se incluye los controles:
 - **pre-insert** : llama al proceso `pck_scheduler_chain_step.create_chain_step` y al proceso `pck_scheduler_chain_rule.define_chain_rule` para crear la condición de ejecución.
 - **pre-delete** : llama al proceso `pck_scheduler_chain_step.drop_chain_step` y al proceso `pck_scheduler_chain_rule.drop_chain_rule` para suprimir la condición de ejecución establecida sobre el paso.

- Parámetros : Este Bloque va sobre la vista `dba_scheduler_program_args`. Se trata de un bloque de solo consulta y funciona como detalle del anterior cuando la acción del paso se trata de un programa.

El botón Establecer condición llamará al formulario “Establecer Condición”.

El Formulario “Establecer Condición” se crea en el mismo formulario que el Gestor de Cadenas pero asignado una Ventana Diferente. El diseño de esta ventana es:

Ilustración 50: Gestor Cadenas - Establecer Condición

Dependiendo del botón en que se llame el título de ella ventana será:

- Establecer Condición Finalización (Si se llama desde el botón de la pestaña “Datos Generales”)
- Establecer Condición Ejecución (Si se llama desde el botón de la pestaña “Pasos y Dependencias”)

Este formulario contiene un Bloque de Datos que no va contra ninguna tabla de base de datos. La lista desplegable Nombre Paso, se cargara de forma dinámica según los pasos que se hayan creado. El



grupo de Radio botones solo estará visible cuando se este estableciendo condiciones de ejecución. Además este formulario no heredará la barra de herramientas.

Los controles disponibles en este formulario son:

- Añadir Condición : Incluye la condición escogida en las condiciones finales
- Aceptar : Incluye las Condiciones en el campo correspondiente, según desde donde se haya llamado a este formulario. Después cierra el Formulario.
- Cancelar : Deshace cualquier cambio y cierra el formulario.

Gestor de Tareas

Al igual que los anteriores gestores, este también incluye el elemento de árbol jerárquico. El **Bloque de Datos** que contiene el Árbol Jerárquico realiza una consulta jerárquica sobre la vista *dba_scheduler_jobs*. Las hojas de este árbol son las Tareas que se han creado. Al seleccionar una hoja se consultarán todos sus datos.

El diseño establecido para este Gestor es:

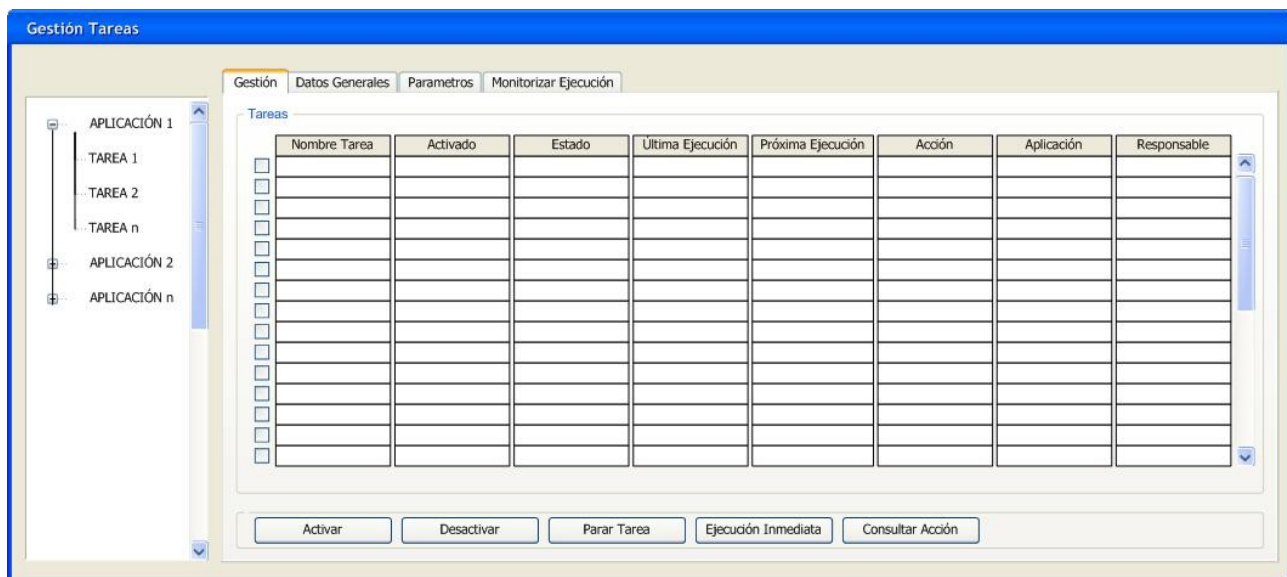


Ilustración 51: Gestor Tareas - Gestión

Esta Pestaña es de solo consulta y contiene un **Bloque de Datos** asociado a la vista *dba_scheduler_jobs*. Al posicionarse sobre un registro de este Bloque se seleccionará la hoja del árbol correspondiente, para ello se incluye el control:

- when-new-record-instance: este control ejecuta código PL/SQL cada vez que se cambia de registro. Para realizar la búsqueda en los nodos del árbol Oracle proporciona el procedimiento FIND_NODE.

Las acciones disponibles en esta pestaña son:

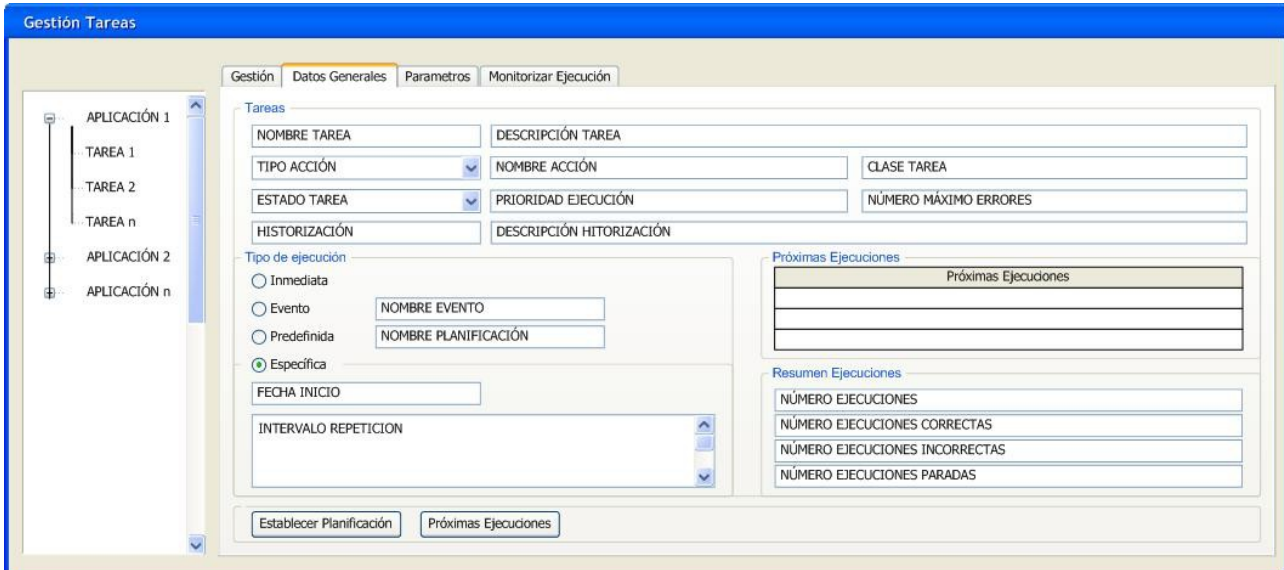
- Activar : llama a *pck_scheduler_comun.enabled*
- Desactivar : llama a *pck_scheduler_comun.disabled*
- Parar : llama a *pck_scheduler_jobs.stop_job*
- Ejecución Inmediatas : llama a *pck_scheduler_jobs.run_job*
- Consultar Acción : según el tipo de acción asociado a la Tarea debe abrir el Gestor de Programas o el Gestor de Cadenas en modo consulta.

La segunda pestaña de "Datos Generales", Contiene un **Bloque de Datos** también asociado a la vista *dba_scheduler_jobs* donde se incluyen los controles:

- **pre-insert** : llama al proceso *pck_scheduler_jobs.create_job*
- **Pre-update** : llama al proceso *pck_scheduler_jobs.set_job_attribute* o *pck_scheduler_comun.set_attribute*

- **pre-delete** : llama al proceso `pck_scheduler_jobs.drop_job`

Se incluye un **Bloque de Datos** que no es de base de datos para evaluar las próximas ejecuciones.

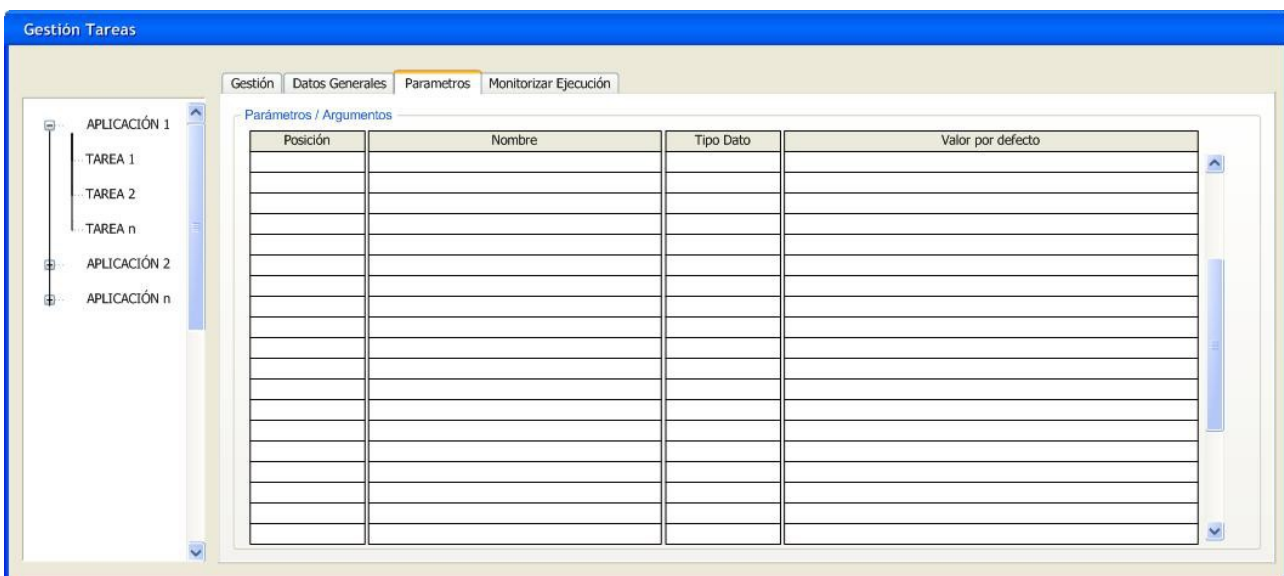


The screenshot shows the 'Gestión Tareas' application interface. The 'Datos Generales' tab is active, displaying various configuration fields for a task. The sidebar on the left shows a tree view of applications and tasks. The main form includes fields for 'NOMBRE TAREA', 'DESCRIPCIÓN TAREA', 'TIPO ACCIÓN', 'NOMBRE ACCIÓN', 'CLASE TAREA', 'ESTADO TAREA', 'PRIORIDAD EJECUCIÓN', 'NÚMERO MÁXIMO ERRORES', 'HISTORIZACIÓN', and 'DESCRIPCIÓN HISTORIZACIÓN'. There are also radio buttons for 'Tipo de ejecución' (Inmediata, Evento, Predefinida, Específica) and input fields for 'NOMBRE EVENTO', 'NOMBRE PLANIFICACIÓN', 'FECHA INICIO', and 'INTERVALO REPETICIÓN'. On the right, there are sections for 'Próximas Ejecuciones' (a table) and 'Resumen Ejecuciones' (fields for 'NÚMERO EJECUCIONES', 'NÚMERO EJECUCIONES CORRECTAS', 'NÚMERO EJECUCIONES INCORRECTAS', and 'NÚMERO EJECUCIONES PARADAS'). Buttons for 'Establecer Planificación' and 'Próximas Ejecuciones' are at the bottom.

Ilustración 52: Gestor Tareas - Datos Generales

Las acciones disponibles en esta pestaña son:

- Tipo Ejecución : Este botón de selección habilita o deshabilita los campos necesarios para al tipo de ejecución que se quiera para la Tarea.
- Establecer Planificación: Este botón esta habilitado solamente para Tipos de Ejecuciones Especificas y realiza la llamada al Formulario de “Detalle ejecución”.
- Próximas Ejecuciones : rellena el **Bloque de Datos** de próximas ejecuciones con los resultados que se reciben al llamar al procedimiento `pck-scheduler_coun.evaluate_calendar_string`. Este botón esta habilitado para los Tipos de Planificaciones Predefinidas y Especificas.



The screenshot shows the 'Gestión Tareas' application interface with the 'Parámetros' tab active. The sidebar on the left is the same as in the previous screenshot. The main area displays a table titled 'Parámetros / Argumentos' with the following columns: 'Posición', 'Nombre', 'Tipo Dato', and 'Valor por defecto'. The table is currently empty.

Ilustración 53: Gestor Tareas - Parámetros

La tercera pestaña “Parámetros” solo esta habilitada cuando el tipo de acción es un Programa. Esta pestaña contiene un **Bloque de Datos** asociado a la vista *dba_scheduler_job_args*. Incluye los siguientes controles:

- **Pre-update** :llama al proceso *pck_scheduler_job_args.set_job_argument_value*.

La última pestaña “Monitorizar Ejecución” esta habilitada cuando el tipo de acción es una cadena y además la tarea esta ejecutándose. Esta pestaña contiene un **Bloque de Datos** asociado a la vista *dba_scheduler_job_run_deatils*. Esta vista contiene el detalle de todas las Tareas que están en ejecución. Al tratarse de Tareas asociadas a una Cadena se visualizan los pasos de dicha cadena y el estado en que se van quedando tras su ejecución.

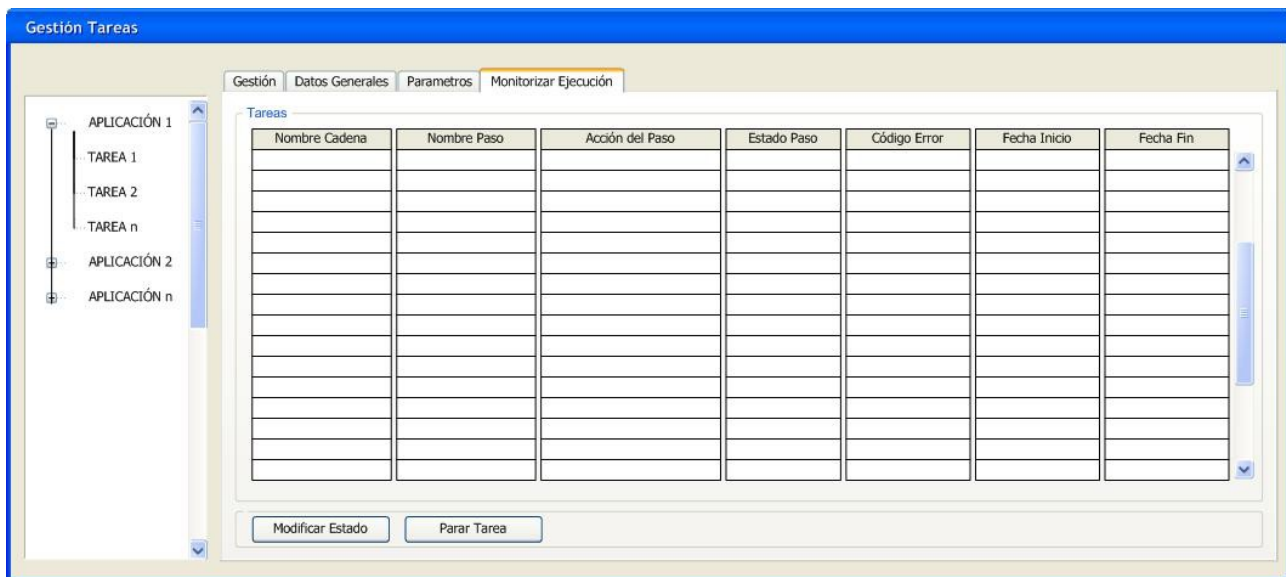


Ilustración 54: Gestor Tareas - Monitorizar Ejecución

Solo se permite dos acciones sobre este **Bloque de Datos**:

- Modificar Estado : llama al procedimiento *pck_scheduler_chains.alter_running_chain*.
- Parar Tarea : llama al procedimiento *pck_scheduler_jobs.stop-job*.

Gestor de Suscripciones

Al igual que los anteriores gestores, este también incluye el elemento de árbol jerárquico. El **Bloque de Datos** que contiene el Árbol Jerárquico realiza una consulta jerárquica sobre la vista *dba_scheduler_jobs*. Las hojas de este árbol son las Tareas que se han creado. Al seleccionar una hoja se consultarán los suscriptores asociados dicha Tarea.

Existen dos formas de visualizar dicha información, se ha diseñado una pestaña para cada una de esas formas:

Ilustración 55: Gestor de Suscriptores - Suscripciones

Esta pestaña de “Suscriptores” visualiza información de todas las tareas en que el usuario esta suscrito. Dicha información se muestra en los siguientes **Bloque de Datos**:

- Datos Suscripción : esta asociado a la vista *dba_job_scheduler_suscriber*, se incluye las siguientes controles:
 - **pre-update** : llama al proceso *pck_scheduler_comun.set_attribute*
 - **pre-delete** : llama al proceso *pck_scheduler_job_suscriber.drop_job_suscriber_email* y elimina el suscriptor de todas las tareas.

- Selección Tareas : esta asociado a la vista *dba_scheduler_jobs*. Este bloque visualiza las tareas que el usuario esta suscrito. Si se selecciona una hoja del árbol, se consultara la tarea en este bloque.

Todas las casillas de control están habilitadas, esto significa que el usuario esta suscrito a dicha tarea. Si se deshabilita la casilla de control se llamará al procedimiento *dba_scheduler_job_suscriber.drop_job_suscriber_email*, quitando así el suscriptor de la tarea. Dejara de visualizarse la tarea en el árbol.

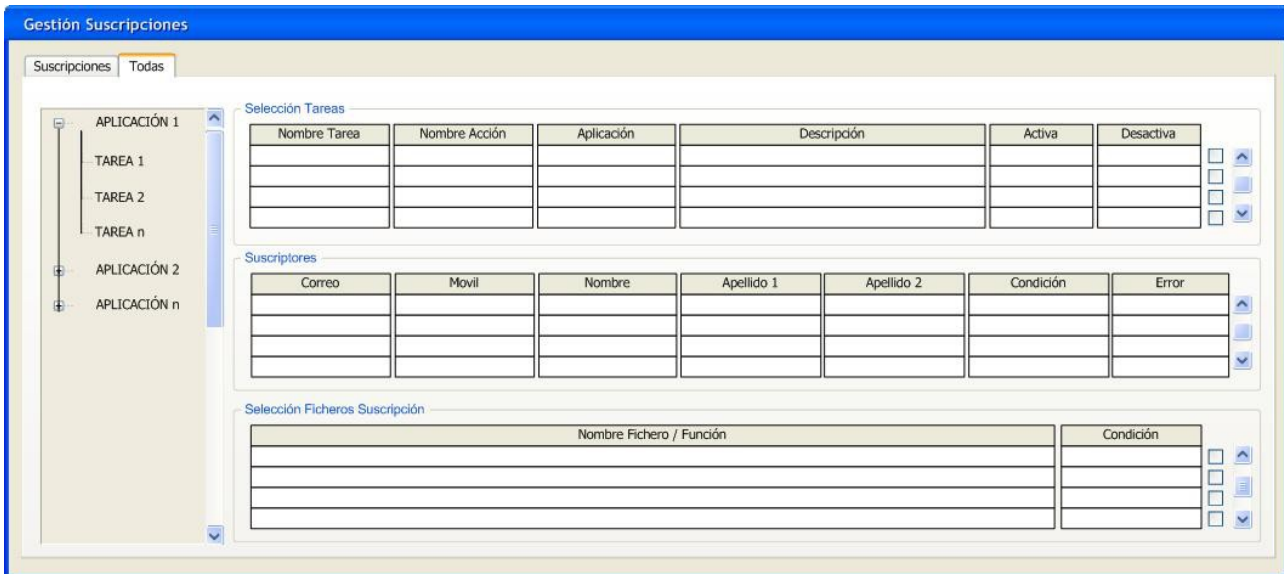
- Selección Ficheros Suscripción: no esta asociado a ninguna vista, por código PL/SQL se rellenará este bloque. Este bloque muestra los resultados del programa asociado a la Tarea; si la Tarea esta asociada a una Cadena, muestra los resultados asociados a los programas asignados a cada paso de la tarea.

La casilla de control se habilitara cuando el usuario esta suscrito al resultado, para ello se utiliza el procedimiento *pck_scheduler_comun.get_attribute*.

Con el mismo procedimiento se sabe la condición de suscripción para el resultado.

Si se deshabilita la casilla de control significa que el usuario ya no quiere recibir mas dicho resultado. Para ello se llamará al procedimiento *pck_scheduler_comun.set_attribute*.

La pestaña de “Todas” visualiza todas las tareas y todos los suscriptores. Muestra la misma información pero ordenada de forma diferente:



The screenshot shows the 'Gestión Suscripciones' application with the 'Todas' tab selected. On the left, there is a tree view showing a hierarchy of 'APLICACIÓN 1', 'TAREA 1', 'TAREA 2', 'TAREA n', 'APLICACIÓN 2', and 'APLICACIÓN n'. The main area contains three data tables:

- Selección Tareas:** A table with columns: Nombre Tarea, Nombre Acción, Aplicación, Descripción, Activa, and Desactiva. It includes checkboxes and arrows for each row.
- Suscriptores:** A table with columns: Correo, Movil, Nombre, Apellido 1, Apellido 2, Condición, and Error. It includes checkboxes and arrows for each row.
- Selección Ficheros Suscripción:** A table with columns: Nombre Fichero / Función and Condición. It includes checkboxes and arrows for each row.

Ilustración 56: Gestor de Suscripciones - Todas

Los **Bloques de Datos** son los siguientes:

- **Selección tareas** : Muestra todas las tareas, a no ser que se seleccione una hoja del árbol, en cuyo caso solo se visualizará dicha Tarea. O las Tareas asociadas a la aplicación seleccionada. No contiene ningún control.
- **Suscriptores** : Este bloque muestra todos los suscriptores asociados a la Tarea seleccionada en el Bloque Anterior. Los controles asociados son:
 - **pre-insert** : llama al proceso *pck_scheduler_job_suscriber.add_job_suscriber_email*.
 - **pre-update** : llama al proceso *pck_scheduler_comun.set_attribute*.
 - **pre-delete** : llama al proceso *pck_scheduler_job_suscriber.drop_job_suscriber_email*.
- **Selección Ficheros Suscripción** : Funciona del mismo modo que se ha explicado en la pestaña anterior.



6 Manual de instalación y usuario

Antes de empezar con los manuales indicar que se debe tener correctamente instalada en el servidor una base de Datos de Oracle y las herramientas necesarias para la ejecución de Formularios.

Dependiendo de configuraciones de base de datos, la instalación puede variar.

6.1 Manual instalación de la aplicación

El proceso de instalación y puesta en marcha de una aplicación en Oracle se deben seguir los siguientes pasos:

- Crear un Esquema de Base de Datos.
- Crear Directorio Servidor para las fuentes.
- Configurar Oracle.

Crear Esquema Aplicación

Se crea el Esquema *SCHEDULER* en la base de datos. Este esquema contendrá los objetos necesarios para el correcto funcionamiento de la aplicación.

Los Objetos a Crear son:

- Tablas
 - schduler_program_result
 - scheduler_job_suscriber
- Vistas
 - dba_scheduler_program_result
 - dba_scheduler_job_suscriber
- Paquetes
 - pck_scheduler_comuns
 - pck_scheduler_windows
 - pck_scheduler_windows_groups
 - pck_scheduler_schedule
 - pck_schedule_program
 - pck_schedule_chains
 - pck_schedule_job_classes
 - pck_scheduler_jobs
- Se crean y asignan estos permisos a todos los objetos del Esquema.
 - scheduler_admin: Permisos de Administración.
 - scheduler_tec: Permisos de técnico

Estos permisos se asignarán al usuario final para diferenciar entre Administrador y Técnico.



Crear Directorio Aplicación

Una vez creados los objetos de base de datos. Se deben crear en el Servidor un directorio para la aplicación que siga la siguiente estructura:

- \...\scheduler
 - \form
 - \bin

El directorio `\scheduler` es el directorio de la aplicación. De él cuelga el directorio `\form` que contendrá las fuentes de la aplicación, es decir, los formularios con extensión `*.fmb`.

Al mismo nivel que el directorio `\form` se encuentra el directorio `\bin` que contendrá las fuentes compiladas en formato `*.fmx`. Este tipo de ficheros es lo que carga el Applet de Java para su visualización.

Una vez creados los directorios se procede a compilar las fuentes `*.fmb` para que genere los `*.fmx` y depurar los errores surgidos debido al cambio de nombre del esquema de base de datos.

Configuración

Una vez se tiene todo compilado correctamente se configura Oracle para que pueda cargar los formularios. Para ello hay que realizar lo siguiente:

- Insertar en la tabla `db_parametros` el directorio de la aplicación, Indicando:
 - Nombre parámetro
 - Descripción
 - Valor (Directorio físico `\...\scheduler`)
- Insertar en la tabla `db_programas` el nombre de las fuentes indicando:
 - Nombre Fuente (Sin extensión)
 - Título
 - Tipo de fuente (FMB)
 - Nombre parámetro donde se encuentra directorio.
 - Carpeta (bin)
 - Fecha Alta: Fecha a partir de la cual el formulario es accesible. Si se deja vacío se entiende que es desde la fecha actual
 - Fecha Baja: Fecha a partir de la cual el formulario dejara de ser accesible. Si se deja vacío se entiende que siempre estara accesible.

Con esto ya esta todo disponible para que los formularios puedan acceder a la aplicación.



6.2 *Manual Usuario*

Con este manual se pretende presentar a los usuarios el funcionamiento de la aplicación de Gestión de Procesos Diferidos. Para ello se tratará los siguientes puntos:

1. Introducción
 - ¿Que se entiende por proceso diferido?
 - Definición de un proceso diferido en Scheduler
2. Opciones de Administrador.
 - Definición
 - Diálogos y Pantallas
3. Gestión del Catalogo de Programas
 - Definición de Programa
 - Diálogos y Pantallas
4. Gestión del Catalogo de Cadenas
 - Definición de Cadena
 - Diálogos y Pantallas
5. Gestión de Tareas
 - Definición de Tarea
 - Diálogos y Pantallas
6. Gestión de Suscripciones
 - Definición de Suscripciones
 - Diálogos y Pantalla

Introducción

El objetivo del nuevo aplicativo SCHEDULER es gestionar la catalogación y planificación de procesos que deben ejecutarse en diferido (Batch).

Se entiende por proceso batch, cualquier objeto ejecutable, que bien por rendimiento, tiempo de cálculo o necesidad de ejecución en un horario determinado, debe ser ejecutado en diferido. En SCHEDULER se pueden definir como procesos Batch cualquiera de los siguientes objetos:

- Procedimientos de Base de Datos³
- Ejecutables (shells)
- Informes
- PL/SQL

¿Qué se entiende por un proceso Batch?

Desde el punto de vista del usuario

Cualquier petición a un Sistema o Aplicación informática interactiva, cuyo tiempo de respuesta en pantalla no sea aceptable; entendiendo como tiempo de respuesta, el tiempo que transcurre entre que se emite la petición y la respuesta a la misma.

Desde el punto de vista del sistema

Toda petición de proceso al sistema, que siendo interactiva, exceda de un tiempo máximo de vida del proceso y suponga un tiempo de respuesta no aceptable.

También toda aquella petición al sistema, que no siendo funcionalmente necesario que tenga una respuesta interactiva inmediata, se estima más eficiente de cara al uso del recurso “ordenador”, que se ejecute en diferido, sin más control por parte del usuario que algún testigo que refleje su estado de ejecución.

Desde el punto de vista de la explotación/operación

Todo aquel proceso que:

- Por su coste en recurso “ordenador”
- Por el impacto que su ejecución pueda generar en el resto del sistema, en especial en los procesos interactivos.
- Por ser resultado de un “ciclo de vida” de la aplicación, resulta conveniente planificar su ejecución en momentos concretos para evitar la concurrencia con la interacción de los usuarios.
- Por no requerir interactividad con el usuario.

Se diseña e implementa para ser ejecutado de forma batch, debidamente planificado en hora y en concurrencia con otras aplicaciones y con relación a otras aplicaciones y procesos batch precedentes.

Definición de un proceso Batch en SCHEDULER

La definición en SCHEDULER de procesos batch, se realizará a través de tareas. Una tarea se define como la asociación entre:

³ Solo se admite la planificación y ejecución de procesos de BD.



- Una acción o componente a ejecutar
- Una planificación u horario de ejecución.
La acción, o componente a ejecutar, podrá ser uno de los siguientes objetos:
- Programa:
 - Puede ser de tipo: PL/SQL, Shell, Informe o Procedimiento almacenado.
- Cadena:
 - Conjunto de programas a ejecutar relacionados entre sí por un conjunto de reglas o dependencias de ejecución.

Opciones de Administración

Definiciones

Clases de Tarea

Una clase de Tarea es un nivel de agrupación de las tareas, proporcionado por SCHEDULER, que permite:

- Asignación de las tareas a grupos de consumo de recursos⁴.
- Definición de un nivel de afinidad (prioridad de ejecución) sobre cierta instancia en una configuración en clúster.
- Configuración de la gestión de los log's de históricos a nivel de clase.
- Priorización de la ejecución de las solicitudes pertenecientes a la misma clase.

Combinando los dos criterios de agrupación y mostrándolos en una tabla se obtiene un mapa de las tareas catalogadas en el Sistema:

		Clases			
		Clase A	Clase B	...	Clase m
Aplicaciones	Apl ₁	P ₁₁ , P ₁₂ , ...	P ₁₃ , ...		
	Apl ₂	-	P ₂₁ , P ₂₂ , ...		P ₂₃ , ...

	Apl _Q	P _{Q1} , P _{Q2} , ...	P _{Q3}	...	P _{Q4}

P_{ij}: Proceso "j" de la aplicación "i"

Ilustración 57: Manual Usuario. Clases de Tarea

Ventanas de Ejecución

Una ventana permite la activación de planes de recursos en diferentes periodos temporales conforme a las necesidades de los administradores del sistema. Las ventanas definen los recursos disponibles para los procesos que se ejecutan mientras éstas estén abiertas.

En la imagen siguiente se muestra un ejemplo de una distribución temporal (ventanas) de un día completo:

00:00	01:00	02:00	03:00	04:00	05:00	06:00	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00
Madrugada						On-line						Tarde				Noche							

Ilustración 58: Manual Usuario. Ventanas de Ejecución

Dado que la definición y alta de ventanas es una tarea de los administradores, este módulo proporcionará únicamente una interfaz de consulta de las que se hayan definido en el planificador de SCHEDULER con propósitos meramente informativos.

⁴ Agrupación de sesiones síncronas o asíncronas con las mismas necesidades de procesamiento.

Planificaciones

Las planificaciones son las codificaciones de los hitos temporales y periódicos definidos en una instalación. Estas codificaciones se podrán asignar durante la planificación de solicitudes y durante la definición y creación de las ventanas⁵.

En la imagen siguiente se muestra un ejemplo de las planificaciones de apertura de las ventanas definidas en el ejemplo anterior.

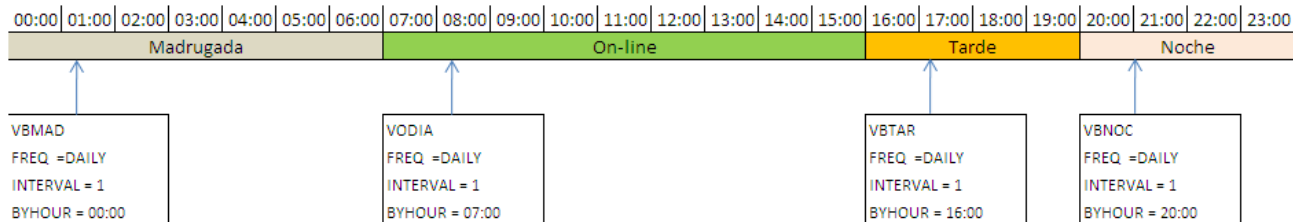


Ilustración 59: Manual Usuario. Planificaciones

Eventos Externos

Un evento es un mensaje informativo enviado a SCHEDULER, por una aplicación externa, para comunicarle cualquier acontecimiento producido en el sistema. La recepción de este mensaje puede desencadenar la ejecución de un proceso específico encargado de gestionar dicho mensaje.

En el esquema siguiente se ilustra el modelo de comunicación de eventos entre aplicaciones externas (incluye procesos planificados) y SCHEDULER.

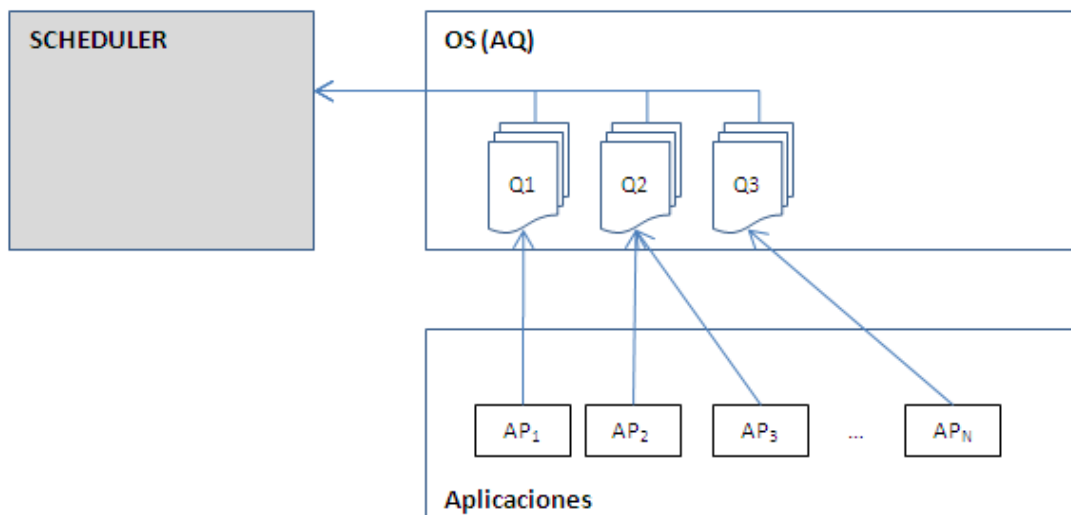


Ilustración 60: Manual Usuario. Eventos Externos

⁵ La creación de ventanas es una tarea para los administradores del sistema tal y como se ha propuesto en el punto anterior.

Diálogos y Pantallas

Mantenimiento de Clases de Tarea

El mantenimiento de *Clases de Tarea*⁶ esta accesible desde la opción *Clases de Tarea* del panel general de la aplicación.



Nombre	Descripción	Consumo recursos	Servicio	Historial	Nivel de log
DEFAULT_JOB_CLASS	This is the default job class.				Ejecución
AUTO_TASKS_JOB_CLASS	System maintenance job class	AUTO_TASK_CONSUMER_GROUP			Ejecución

Ilustración 61: Manual Usuario. Mantenimiento Clases de Tarea

A continuación se describen los campos asociados a una Clase de Tarea que será informado en esta opción junto con sus opciones de obligatoriedad y editabilidad.

Clases de Tarea			
Campo	Descripción / Comentarios	Oblig.	Edit.
Nombre	Identificador de la Clase de Tarea.	S	N
Descripción	Descripción de la Clase de Tarea.	S	S
Consumo Recursos	<i>Grupo de Consumo de Recurso</i> asociado a la Clase de Tarea. Un <i>Grupo de Consumo de Recurso</i> es un conjunto de sesiones síncronas o asíncronas agrupadas en función de las necesidades del proceso. Es excluyente del campo <i>Servicio</i> .	N	S
Servicio	Servicio de BD a la cual las Tareas asociadas a esta Clase se asocian. Es excluyente del campo <i>Consumo Recursos</i>	N	S
Historial	Número de días que permanecen los registros del log en el sistema antes de ser eliminados.	S	S
Nivel de Log	Nivel de log que se registra. Existen dos opciones: <ul style="list-style-type: none"> LOG EJECUCIÓN: Solo se registra la información de la ejecución. LOG_COMPLETO: Se registra todas las operaciones de las Tareas asociadas a esta <i>Clase de Tarea</i>. 	S	S

Tabla 36: Manual usuario. Clases de Tarea

Alta de Clases de Tarea

Para realizar el alta de una *Clase de Tarea* se deben cumplir las siguientes validaciones:

- Deben estar informados todos los campos obligatorios.
- No debe existir otro objeto con el mismo nombre.
- Debe estar informado el campo *Consumo Recursos* o *Servicio*.

El alta de una Clase de Tarea desencadena la siguiente acción:

⁶ En las aplicación final el concepto de Tarea se ha traducido a Solicitud.



- Alta de la *Clase de Tarea* en la estructura de SCHEDULER.

Modificación Clases de Tarea

Para realizar la modificación de una *Clase de Tarea* deben cumplirse las siguientes validaciones:

- Deben estar informados todos los campos obligatorios.
- No debe existir otro objeto con el mismo nombre.
- Debe estar informado el campo *Consumo Recursos* o *Servicio*.

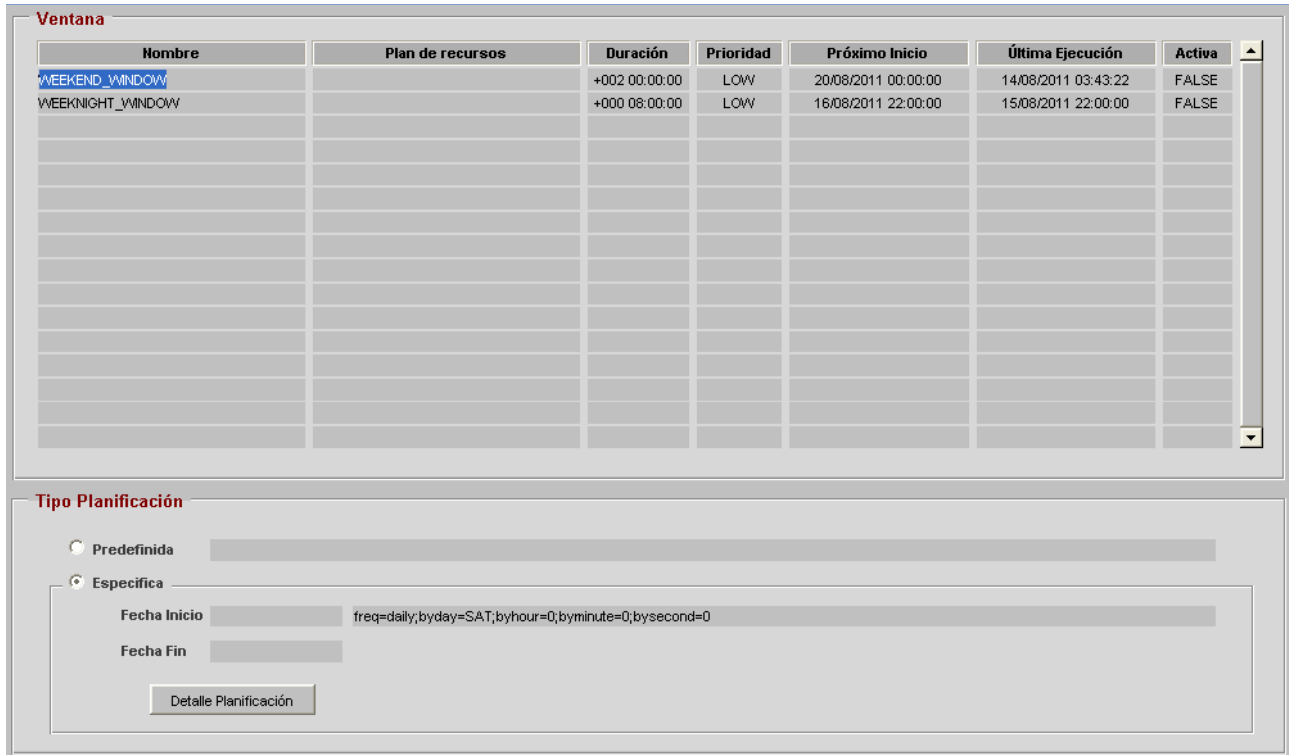
Baja Clases de Tarea

Para realizar la baja de una Clase de Tarea deben cumplirse las siguientes validaciones:

- No debe existir una Tarea activa que tenga como referencia la Clase de Tarea.

Consulta de Ventanas de Ejecución

La Consulta de *Ventanas de Ejecución* esta accesible desde la opción *Ventanas de Ejecución* del panel general de la aplicación.



The screenshot displays a software interface for managing execution windows. It is divided into two main sections: a table and a configuration panel.

Ventana

Nombre	Plan de recursos	Duración	Prioridad	Próximo Inicio	Última Ejecución	Activa
WEEKEND_WINDOW		+002 00:00:00	LOW	20/08/2011 00:00:00	14/08/2011 03:43:22	FALSE
WEEKNIGHT_WINDOW		+000 08:00:00	LOW	16/08/2011 22:00:00	15/08/2011 22:00:00	FALSE

Tipo Planificación

Predefinida

Específica

Fecha Inicio: freq=daily;byday=SAT;byhour=0;byminute=0;bysecond=0

Fecha Fin:

Ilustración 62: Manual Usuario. Consulta Ventanas de Ejecución

Dado que la definición y alta de ventanas es una tarea de los administradores, este módulo proporcionará únicamente una interfaz de consulta de las que se hayan definido en el planificador de SCHEDULER con propósitos meramente informativos.

A continuación se describen los campos asociados a una ventana que estará disponibles en esta opción junto con sus condiciones de obligatoriedad y editabilidad.

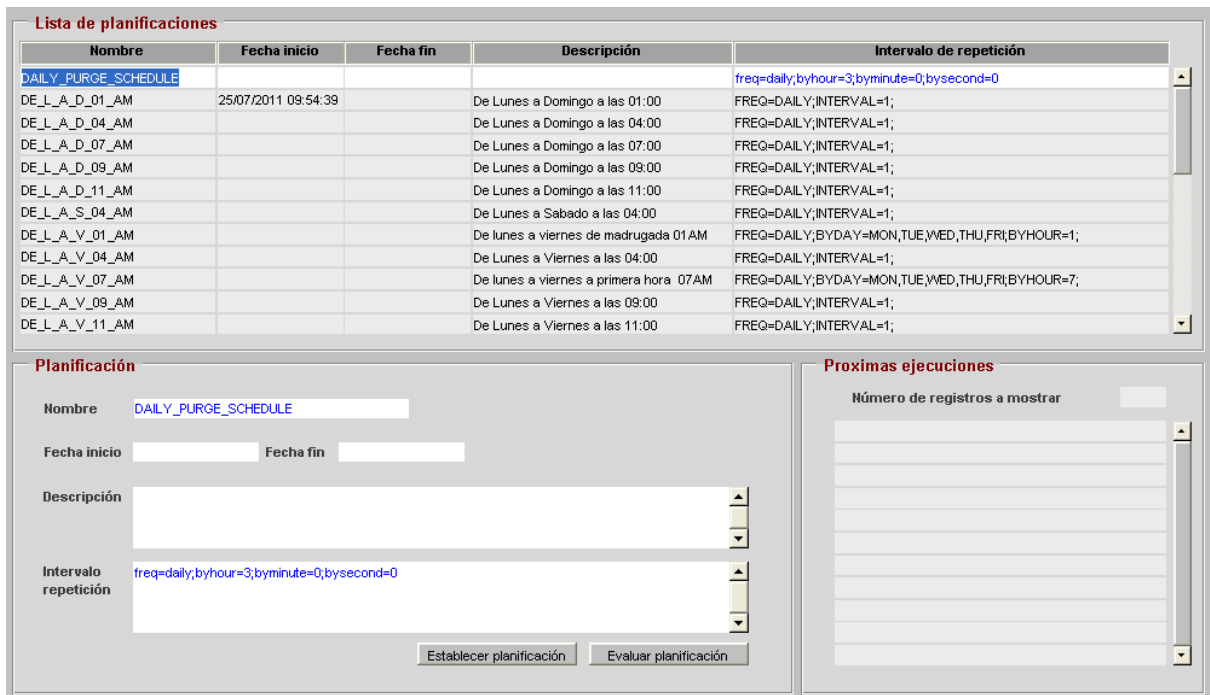


Ventanas Ejecución			
Campo	Descripción / Comentarios	Oblig.	Edit.
Nombre	Identificación de la ventana de ejecución.	N	N
Plan de recurso	Plan de recurso que se activa cuando se inicia la ventana.	N	N
Duración	Tiempo que permanece la ventana abierta.	N	N
Prioridad	En caso de superposición de ventanas. Este campo indicará cual es la mas prioritaria.	N	N
Próximo inicio	Indica la próxima ocasión que la ventana será abierta.	N	N
Última ejecución	Indica la última vez que la ventana fue abierta.	N	N
Activa	Indica si la Ventana esta Activa o Desactiva	N	N
Planificación	(Predefinida) Planificación asociada	N	N
Fecha Inicio	(Específica) Fecha Inicio de la Ventana	N	N
Fecha Fin	(Específica) Fecha fin de la Ventana	N	N
Intervalo Repetición	(Específica) Condiciones de apertura para la ventana.	N	N

Tabla 37: Manual Usuario. Consulta Ventanas Ejecución.

Mantenimiento Planificaciones

El *Mantenimiento de Planificaciones* esta accesible desde la opción *Mantenimiento de Planificaciones* del panel general de la aplicación.



Nombre	Fecha inicio	Fecha fin	Descripción	Intervalo de repetición
DAILY_PURGE_SCHEDULE				freq=daily;byhour=3;byminute=0;bysecond=0
DE_L_A_D_01_AM	25/07/2011 09:54:39		De Lunes a Domingo a las 01:00	FREQ=DAILY;INTERVAL=1;
DE_L_A_D_04_AM			De Lunes a Domingo a las 04:00	FREQ=DAILY;INTERVAL=1;
DE_L_A_D_07_AM			De Lunes a Domingo a las 07:00	FREQ=DAILY;INTERVAL=1;
DE_L_A_D_09_AM			De Lunes a Domingo a las 09:00	FREQ=DAILY;INTERVAL=1;
DE_L_A_D_11_AM			De Lunes a Domingo a las 11:00	FREQ=DAILY;INTERVAL=1;
DE_L_A_S_04_AM			De Lunes a Sabado a las 04:00	FREQ=DAILY;INTERVAL=1;
DE_L_A_V_01_AM			De lunes a viernes de madrugada 01AM	FREQ=DAILY;BYDAY=MON,TUE,WED,THU,FRI;BYHOUR=1;
DE_L_A_V_04_AM			De Lunes a Viernes a las 04:00	FREQ=DAILY;INTERVAL=1;
DE_L_A_V_07_AM			De lunes a viernes a primera hora 07AM	FREQ=DAILY;BYDAY=MON,TUE,WED,THU,FRI;BYHOUR=7;
DE_L_A_V_09_AM			De Lunes a Viernes a las 09:00	FREQ=DAILY;INTERVAL=1;
DE_L_A_V_11_AM			De Lunes a Viernes a las 11:00	FREQ=DAILY;INTERVAL=1;

Ilustración 63: Manual Usuario. Mantenimiento Planificaciones

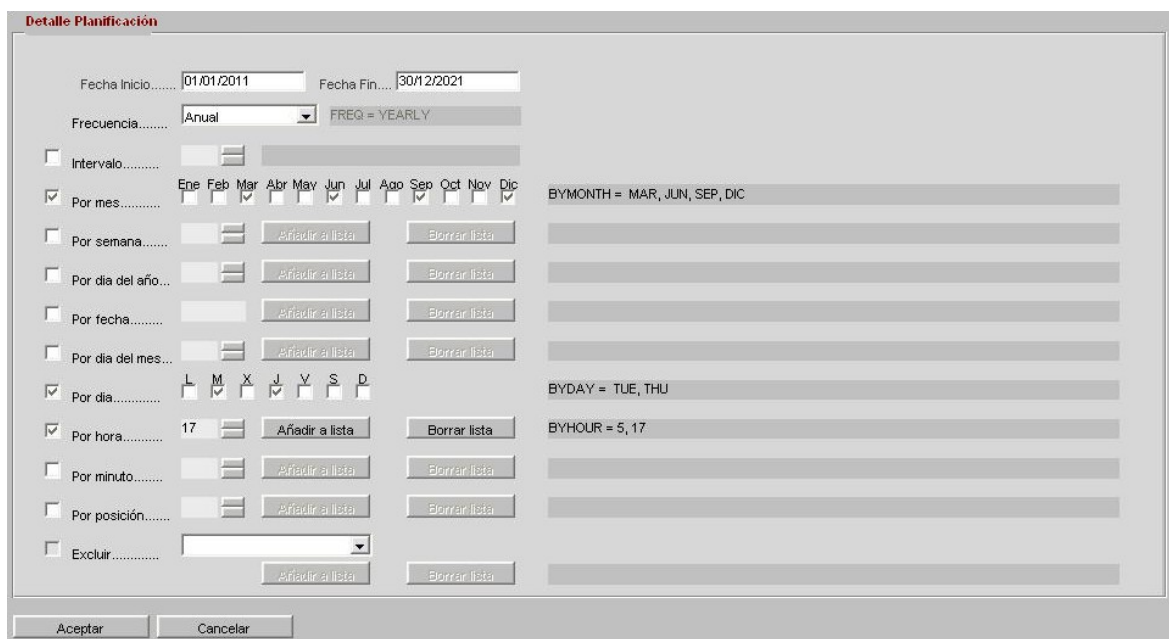


Ilustración 64: Manual Usuario. Detalle Planificación



A continuación se describen los campos asociados a una planificación que se informan en esta opción junto con sus condiciones de obligatoriedad y editabilidad.

Planificación			
Campo	Descripción / Comentarios	Oblig.	Edit.
Nombre	Identificación de la planificación.	S	N
Descripción	Descripción de la planificación	N	S
Fecha Inicio	Fecha a partir de la cual la planificación esta activa.	N	S
Fecha Fin	Fecha a partir de la cual la planificación deja de estar activa.	N	S
Intervalo Repetición	Define la periodicidad de la planificación: <ul style="list-style-type: none">Frecuencia (Diario, Semanal, Mensual...)Cláusulas adicionales Especificación para la Frecuencia. (Número de repeticiones, días de la semana, días del mes, horas,...) Esta información se puede introducir de forma manual utilizando el campo habilitado para ello, o utilizar el formulario de <i>Detalle planificación</i> , que facilita la introducción de datos si se desconoce la sintaxis de SCHEDULER..	S	S

Tabla 38: Manual Usuario. PPlanificación

Alta de Planificación

Para realizar la alta de una planificación deben cumplirse las siguientes validaciones:

- Deben estar informados todos los campos obligatorios.
- No debe existir otro Objeto de SCHEDULER con el mismo nombre.

Modificación Planificación

Para realizar la modificación de una planificación deben cumplirse las siguientes validaciones:

- Deben estar informados todos los campos obligatorios.

Baja Planificación

Para realizar la baja de una planificación deben cumplirse las siguientes validaciones:

- No debe existir ninguna Ventana o Tarea asociada a la planificación.

Evaluar Planificación

Esta funcionalidad consultar las próximas fechas de ejecución, para así, validar la correcta definición de la planificación.

La evaluación de la planificación desencadena las siguientes acciones:

- Rellena la Tabla con las fechas de las próximas ejecuciones.
- El número de fechas mostradas se indica en el campo *Número de Registros a Mostrar* (Valor por defecto 10).

Mantenimiento de Eventos Externos

El *Mantenimiento de Eventos Externos* esta accesible desde la opción *Mantenimiento de Eventos Externos* del panel general de la aplicación.

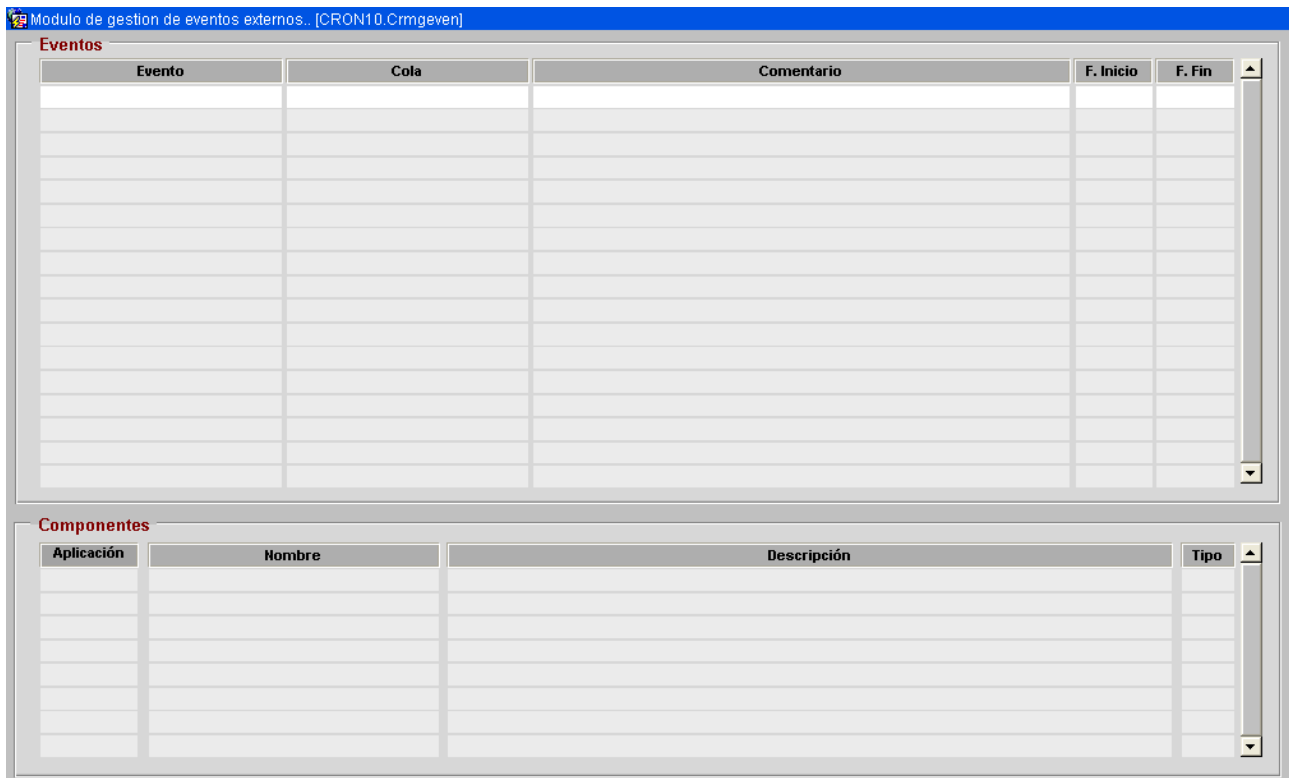


Ilustración 65: Manual Usuario. Mantenimiento Eventos Externos

Los campos asociados a un evento se especifica en la siguiente tabla junto con sus condiciones de obligatoriedad y editabilidad:

Eventos Externos			
Campo	Descripción / Comentarios	Oblig.	Edit.
Nombre	Identificación del Evento Externo	S	N
Descripción	Descripción del Evento Externo	N	S
Cola	Nombre de la Cola donde se incluyen los mensajes de las aplicaciones externas.	S	S
Condición	Expresión lógica que determina cuando se produce el Evento.	S	S
Fecha inicio	Fecha a partir de la cual el evento empieza a estar activo.	N	S
Fecha fin	Fecha a partir de la cual el evento deja de estar activo.	N	S

Tabla 39: Manual Usuario. Eventos Externos

Las aplicaciones externas con permiso para generar el evento se especifica en el marco de



Componentes⁷. Dicho marco es un detalle del Evento Externo actualmente posicionado. Los campos asociados a los Componentes se especifica en la siguiente tabla junto con sus condiciones de obligatoriedad y editabilidad:

Componentes			
Campo	Descripción / Comentarios	Oblig.	Edit.
Aplicación	Identificador de la Aplicación	S	N
Nombre	Nombre del Componente de la Aplicación que va a utilizar el evento	S	N
Descripción	Descripción del Componente	N	N
Tipo	Tipo de Componente	N	N

Tabla 40: Manual Usuario. Eventos Externos. Componentes

Alta de Evento

Para realizar el alta de un Evento Externo debe cumplirse las siguientes validaciones:

- Deben informarse todos los campos obligatorios
- No debe existir otro Objeto SCHEDULER con el mismo nombre

Modificación de Evento

Para realizar la modificación de un Evento Externo debe cumplirse las siguientes validaciones:

- No debe estar asociado a ninguna Tarea o Cadena.

Baja de Evento

Para realizar la baja de un Evento Externo debe cumplirse las siguientes validaciones:

- No debe estar asociado a ninguna Tarea o Cadena.

Generar Evento

Para facilitar la Generación de Eventos se proporciona la siguiente función de base de datos.

FU_GENERAR_EVENTO	
Función genérica para la generación de eventos	
Parámetros Entrada	Aplicación Identificador de la aplicación que va a generar el evento. Componente Nombre del componente que va a generar el evento. Evento Identificador del Evento que se va a generar. Mensaje Mensaje que se quiere introducir en el Evento.
Lógica	Valida que el Componente esta asociado al Evento. En caso afirmativo genera el Evento. En caso negativo devolverá un mensaje de error indicando esta situación.

⁷ Entiéndase como Componente cualquiera de los siguientes objetos que se definen para una aplicación: Formulario, Procedimiento o Función de Base de Datos.



Gestión Programas

Definición de Programa

Programa

Un programa define la estructura de un componente ejecutable encargado de realizar una acción cuando se planifique su ejecución a través de una Cadena o Tarea.

Los tipos de programa que se pueden catalogar para su posterior planificación son:

- Procedimientos Almacenados de Base de Datos
- Programas Externos (Shells)
- Informes
- Código PL/SQL

Las características que se asocian a un programa son:

- Aplicación
- Ejecutable
- Parámetros
- Resultado
- Responsable

Parámetros

Como parte de la definición de un programa, es necesario establecer los parámetros que debe recibir al ser invocado.

Solo se admiten parámetros de entrada pertenecientes a alguno de los siguientes tipos:

- Carácter
- Numérico
- Fecha

Resultados

Un programa puede generar resultados, dicho resultado se genera en forma de fichero. La asociación de Ficheros de Resultado a un programa permite que se puedan remitir dichos ficheros por notificación al finalizar cada ejecución del programa.

Los ficheros de resultados pueden ser de dos tipos:

- IN – Informe : Informe, Hoja de Cálculo o cualquier otro fichero que se genere como resultado tras la ejecución del programa.
- LE - Log de Ejecución : Fichero asociado al resultado de ejecución del programa.

En programas de tipo Ejecutable Externo, se genera por defecto dos resultados de tipo LE, correspondientes a las salida estándar y error del ejecutable.

En programas de tipo Informe, se generará por defecto un resultado de tipo IN, asociado a la salida de la ejecución del informe, y un resultado de tipo LE asociado al log de ejecución del Informe.

Diálogos y Pantallas

El *Mantenimiento de Programas* esta accesible desde la opción *Gestión de Procesos*⁸ del panel general de la aplicación.

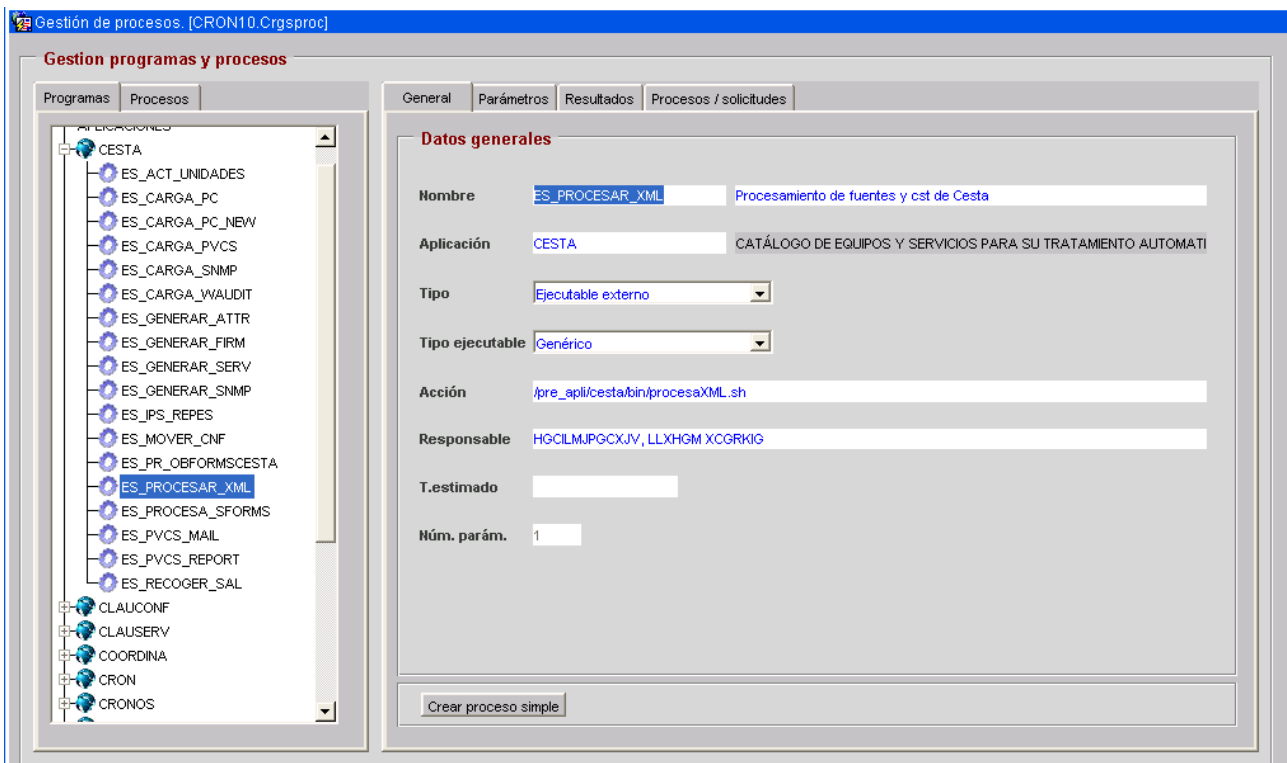


Ilustración 66: Manual Usuario. Gestión Programas.

Este formulario permite la navegación entre los distintos programas de una aplicación. Para cada programa estará disponibles las siguientes opciones, en las distintas pestañas de este formulario:

- Datos Generales : Atributos específicos del programa.
- Mantenimiento de parámetros : Parámetros o Argumentos asociados al programa si los tiene.
- Mantenimiento de resultados : Ficheros de Resultados generados por el programa.
- Procesos / Solicitudes⁹ : Opción de consulta para visualizar las Cadenas y Tareas en los que se encuentra referenciado el programa.

8 Gestión de Procesos: En esta opción de menú unifica el mantenimiento de Programas y Cadenas. Las Cadenas en esta aplicación se han definido como Procesos.

9 Solicitudes: Es el equivalente a las Tareas.

Datos Generales

El mantenimiento de los datos específicos de un programa se realiza desde la pestaña Datos Generales. A continuación se describen los campos asociados al programa junto con sus opciones de obligatoriedad y editabilidad:

Datos Generales			
Campo	Descripción / Comentarios	Oblig.	Edit.
Aplicación	Identificador de la aplicación al que pertenece el Programa.	S	S
Nombre	Identificación del Programa	S	N
Descripción	Descripción del Programa	N	S
Tipo	Tipo de programa: <ul style="list-style-type: none"> • Procedimiento Almacenado • Ejecutable Externo • PL/SQL 	S	N
Tipo Ejecutable	Opción Disponible si el Tipo es: Ejecutable Externo. Tipo de ejecutable externo disponibles: <ul style="list-style-type: none"> • Genérico: Ejecución de cualquier shell. • Informe: Ejecución de informes. 	S	N
Acción	Acción que realiza el programa: <ul style="list-style-type: none"> • Procedimiento Almacenado: Permite informar cualquier proceso de base de datos. • Ejecutable Externo Genérico: Se debe indicar Directorio y el nombre del Ejecutable. • Ejecutable Externo Informe: Se informa de forma automática • PL/SQL: Permite introducir código PL/SQL. 	S	N
Informe	Opción Disponible si Tipo Ejecutable es: Informe Nombre del Informe	S	N
Número Parámetros	Número parámetros que recibe la Acción	S	N
Tiempo Ejecución	Duración de la ejecución expresado en HH:MM:SS	S	S
Responsable	Usuario encargado del Mantenimiento del programa.	S	S

Tabla 41: Manual Usuario. Gestión Programas. Datos Generales

Alta de Programa

Para realizar el alta de un programa deben cumplirse las siguientes validaciones:

- Campos Obligatorios informados.
- No debe existir un objeto SCHEDULER con el mismo nombre.
- Debe existir la aplicación a la que hace referencia.
- Debe existir el responsable introducido.



El Alta de un programa desencadena las siguientes acciones:

- Alta de programa en estructura de SCHEDULER.
- Si Tipo Programa es Procedimiento Almacenado
 - Se recupera automáticamente los parámetros asociados al Programa
- Si Tipo Programa es Ejecutable Externo Genérico, se insertan automáticamente los siguientes resultados:
 - LE: Resultado con nombre "nombre_shell.log". Este resultado referencia a la salida estándar de la ejecución del shell.
 - LE: Resultado con nombre "nombre_shell.err". Este resultado referencia a la salida de error de la ejecución del shell.
- Si Tipo Programa es Ejecutable Externo Informe
 - Se insertan automáticamente los siguientes parámetros:
 - REPORT: Nombre del informe a ejecutar.
 - DESNAME: nombre del fichero de salida que se genera.
 - Se insertan automáticamente los siguientes resultados:
 - LE: Resultado con nombre "nombre_informe.log". Este resultado referencia a la salida estándar de la ejecución del informe.
 - IN: Resultado con nombre "nombre_informe.pdf". Este resultado referencia al informe de salida generado.

Modificación Programa

Para realizar la modificación del programa deben cumplirse las siguientes validaciones:

- Debe existir la aplicación a la que hace referencia.
- Debe existir el responsable introducido.
- No se permite modificar el campo Número de Parámetros. Si el programa esta referenciado en una Cadena o Tarea.

Baja Programa

Para realizar la baja del programa deben cumplirse las siguientes validaciones:

- No se permite modificar el campo Número de Parámetros. Si el programa esta referenciado en una Cadena o Tarea.

Creación Proceso Simple

Esta funcionalidad permite crear una Cadena de forma automática, asociando el Programa actualmente posicionado, tan solo hay que informar el nombre de la Cadena.

Para crear la Cadena deben cumplirse las siguientes validaciones:

- El nombre especificado para la Cadena no debe existir en SCHEDULER.

Mantenimiento de Parámetros

El mantenimiento de los parámetros asociados a un programa se realiza desde la pestaña *Parámetros*:

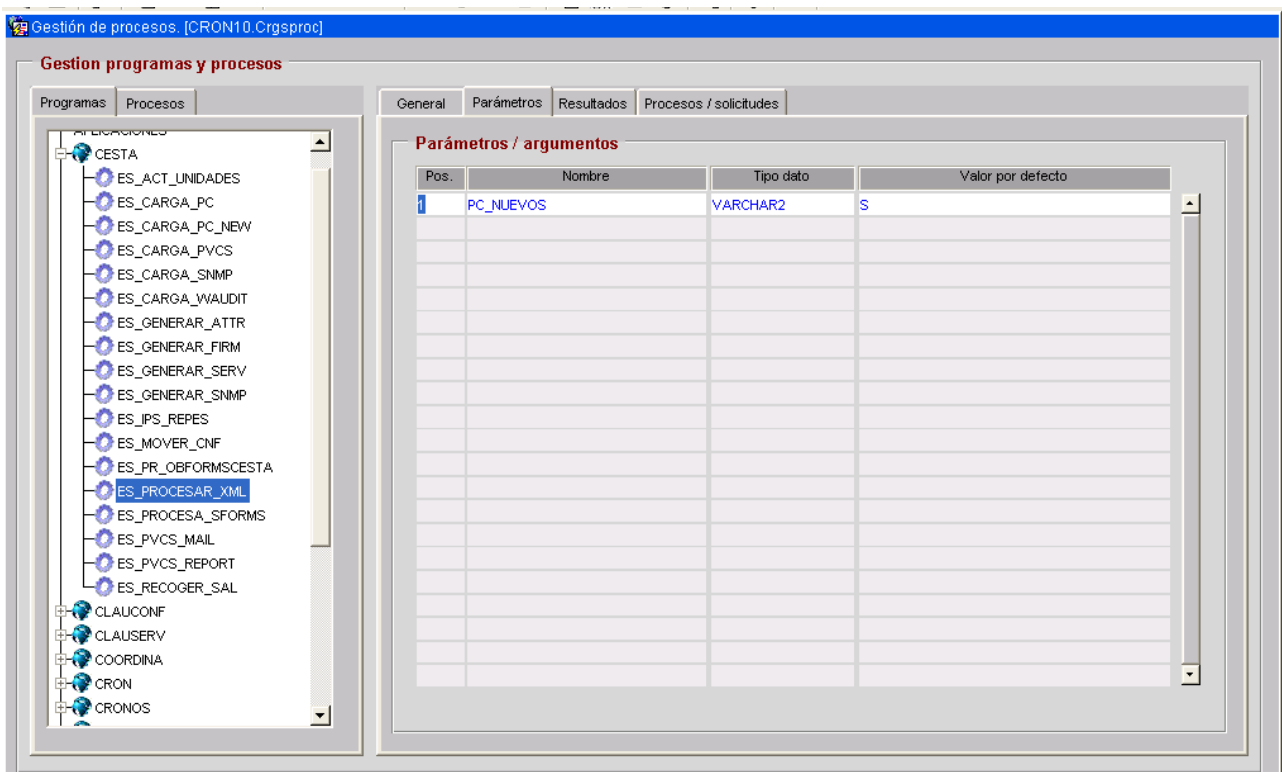


Ilustración 67: Manual Usuario. Gestión Programas. Parámetros

A continuación se describen los campos asociados a un parámetro que serán informados en esta opción junto con sus condiciones de obligatoriedad y editabilidad.

Parámetros Programa			
Campo	Descripción / Comentarios	Oblig.	Edit.
Posición	Posición del parámetro en el programa	S	N
Nombre	Nombre del parámetro	S	S
Tipo	Tipo de dato: <ul style="list-style-type: none"> Numérico Carácter Fecha 	S	S
Valor por defecto	Valor por defecto asociado al parámetro	N	S

Tabla 42: Manual Usuario. Gestión Programas. Parámetros



Alta de Parámetro

Para realizar el alta de un parámetro se deben cumplir las siguientes validaciones:

- El Programa debe ser de Tipo Ejecutable Externo¹⁰.
- No puede existir otro parámetro asociado al mismo programa, en la misma posición ni con el mismo nombre.
- Solo se permite dar de alta los números de parámetros que se han indicado en los Datos Generales del programa.
- Si el Tipo de Programa es un Ejecutable Externo de Tipo Informe los dos primeros parámetros son fijos.

Modificación Parámetro

Para realizar la modificación de un parámetro se deben cumplir las siguientes validaciones:

- No puede existir otro parámetro asociado al mismo programa, en la misma posición ni con el mismo nombre.
- Si el Programa esta referenciado en una Cadena o Tarea solo se permite modificar el campo Valor por Defecto.
- Si el Tipo de Programa es un Ejecutable Externo de Tipo Informe los dos primeros parámetros son fijos. Permitiendo solo modificar el campo Valor por Defecto.

Baja de Parámetro

Para realizar la baja de un parámetro se deben cumplir las siguientes validaciones:

- El Programa no puede estar referenciado en una Cadena o Tarea.
- Si el Tipo de Programa es un Ejecutable Externo de Tipo Informe los dos primeros parámetros son fijos, es decir no se pueden eliminar.

¹⁰ Para Procedimientos Almacenados SCHEDULER rellena automáticamente los parámetros.

Mantenimiento de Resultados

El mantenimiento de los resultados asociados a un programa se realiza desde la pestaña *Resultados*:

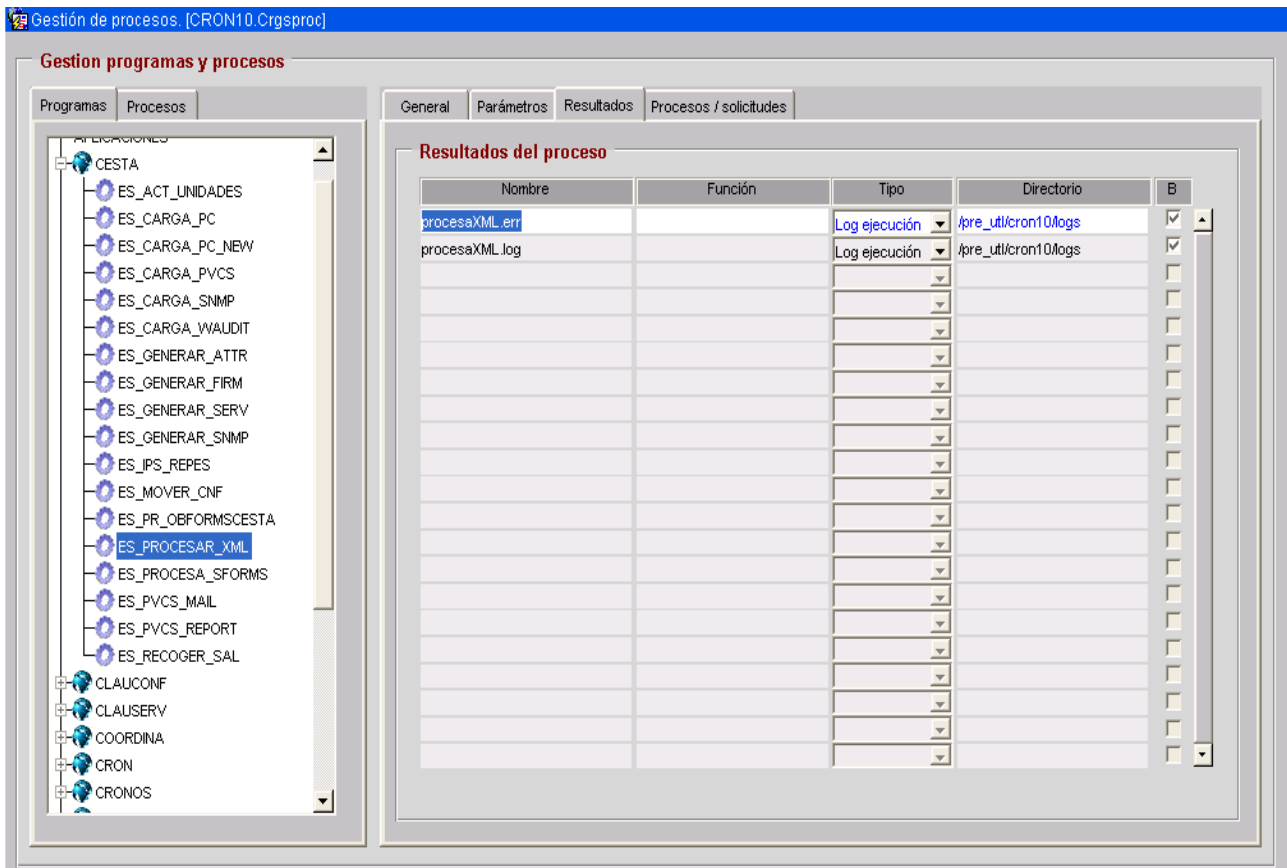


Ilustración 68: Manual Usuario. Gestión Programas. Resultados.

A continuación se describen los campos asociados a un resultado que serán informados en esta opción junto con sus condiciones de obligatoriedad y editabilidad.

Resultados Programa			
Campo	Descripción / Comentarios	Oblig.	Edit.
Nombre	Nombre del fichero.	N	S
Función	Función que determina el nombre del fichero	N	S
Tipo	Tipo Resultado: <ul style="list-style-type: none"> Informe Log Ejecución 	S	S
Directorio	Directorio donde se ubica el resultado	S	S
Borrado	Indicador de Borrado tras ser enviado la notificación.	S	S

Tabla 43: Manual Usuario. Gestión Programas. Resultados.



Alta de Resultado

Para realizar la alta de resultados deben cumplirse las siguientes validaciones:

- Campos Obligatorios informados.
- Debe Informarse el Nombre del fichero o la Función que obtiene el nombre.

Modificación de Resultado

Para realizar la modificación de resultados deben cumplirse las siguientes validaciones:

- Campos Obligatorios informados.
- Debe Informarse el Nombre del fichero o la Función que obtiene el nombre.

Baja de Resultado

Para realizar la baja de resultados deben cumplirse las siguientes validaciones:

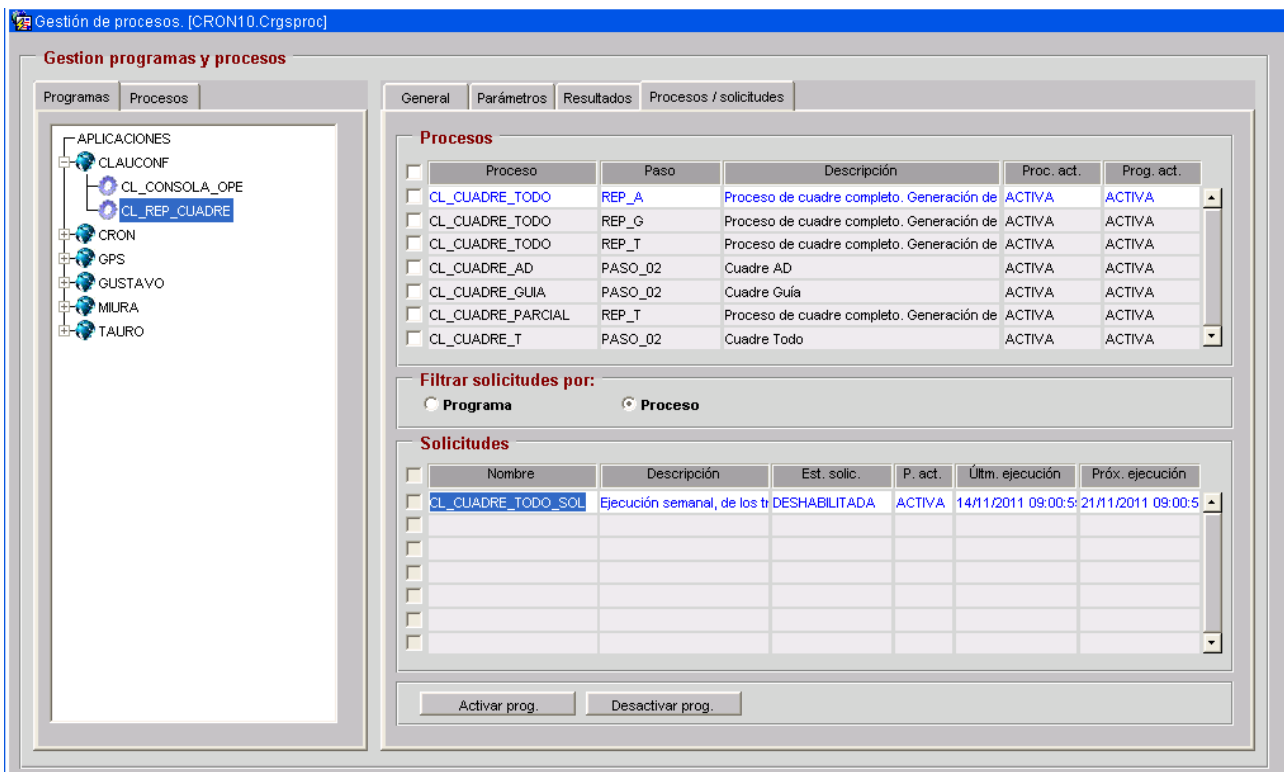
- No debe existir ninguna suscripción al fichero.

Consulta de Cadenas y Tareas

Esta Opción permite la consulta de objetos, Cadenas y Tareas, en los que se encuentra referenciado el programa. Se muestra la siguiente información:

- Cadenas en las que el Programa esta referenciado.
- Según la selección que realice el usuario:
 - Tareas a las que se encuentre referenciado la Cadena seleccionada.
 - Tareas a las que se encuentre referenciado el Programa.

La Consulta de Cadenas y tareas asociados a un programa se realiza desde la pestaña *Procesos / Solicitudes*¹¹:



The screenshot shows the 'Gestión de procesos' application window. The main area is divided into two sections: 'Procesos' and 'Solicitudes'. The 'Procesos' section contains a table with columns: Proceso, Paso, Descripción, Proc. act., and Prog. act. The 'Solicitudes' section contains a table with columns: Nombre, Descripción, Est. solíc., P. act., Últm. ejecución, and Próx. ejecución. Below the tables are buttons for 'Activar prog.' and 'Desactivar prog.'

Proceso	Paso	Descripción	Proc. act.	Prog. act.
<input type="checkbox"/> CL_CUADRE_TODO	REP_A	Proceso de cuadro completo. Generación de	ACTIVA	ACTIVA
<input type="checkbox"/> CL_CUADRE_TODO	REP_G	Proceso de cuadro completo. Generación de	ACTIVA	ACTIVA
<input type="checkbox"/> CL_CUADRE_TODO	REP_T	Proceso de cuadro completo. Generación de	ACTIVA	ACTIVA
<input type="checkbox"/> CL_CUADRE_AD	PASO_02	Cuadre AD	ACTIVA	ACTIVA
<input type="checkbox"/> CL_CUADRE_GUIA	PASO_02	Cuadre Guía	ACTIVA	ACTIVA
<input type="checkbox"/> CL_CUADRE_PARCIAL	REP_T	Proceso de cuadro completo. Generación de	ACTIVA	ACTIVA
<input type="checkbox"/> CL_CUADRE_T	PASO_02	Cuadre Todo	ACTIVA	ACTIVA

Nombre	Descripción	Est. solíc.	P. act.	Últm. ejecución	Próx. ejecución
<input type="checkbox"/> CL_CUADRE_TODO_SOL	Ejecución semanal, de los tr	DESHABILITADA	ACTIVA	14/11/2011 09:00:5	21/11/2011 09:00:5

Ilustración 69: Manual Usuario. Gestión Programas. Consulta Cadenas y Tareas

Desde esta opción se permite las siguientes acciones Activar y Desactivar el programa.

Activar Programa

Permite la Activación del programa actual en cada uno de los Objetos referenciados.

Desactivar Programa

Permite la des-activación del programa actual en cada uno de los Objetos referenciados.

¹¹ Procesos: Cadenas. Solicitudes: Tareas.

Gestión de Cadenas

Definición de Cadenas

Cadena

Una cadena es una agrupación de acciones (programas, eventos y otras Cadenass) con dependencias entre ellas, y una lógica de aplicación de las mismas.

Como parte de la definición de una Cadena, será necesario establecer los pasos que componen el mismo junto con el conjunto de condiciones que deben cumplirse para iniciar la ejecución de cada paso.

Pasos y Dependencias

Definición de pasos

Un paso puede hacer referencia a un programa, un evento o a otra Cadena que deberán estar previamente catalogados en el sistema.

- Un paso que haga referencia a un programa ejecutará dicho programa al iniciarse el paso. Se podrán modificar los valores de entrada de los parámetros del programa para la ejecución del mismo en el contexto específico de este paso.
- Un paso que haga referencia a una Cadena ejecutará dicha Cadena con los parámetros que tenga asociados en la definición de cada uno de sus pasos.
- Un paso que haga referencia a un evento, durante la ejecución de la Cadena quedará a la espera de que se produzca dicho evento para completarse. Este tipo de pasos, será de utilidad para establecer dependencias con eventos externos a la Cadena que se está ejecutando.

En el esquema siguiente se muestran los estados en los que se puede encontrar un paso:

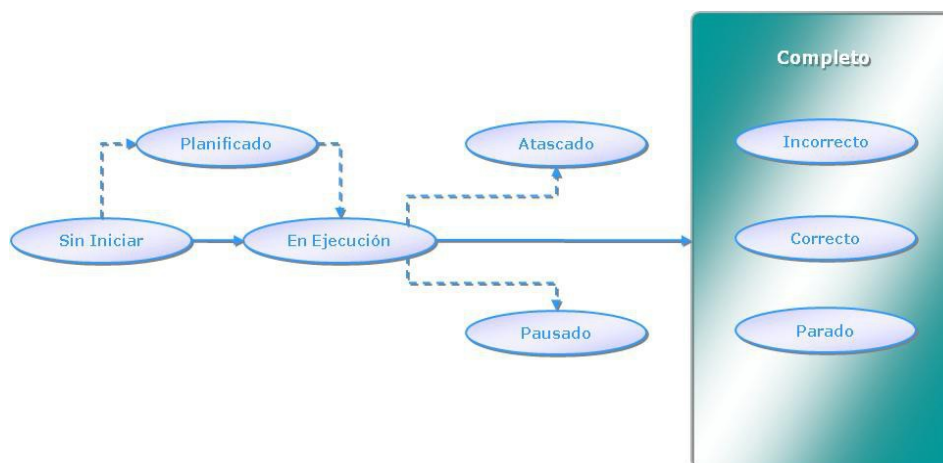


Ilustración 70: Manual Usuario. Gestión Cadenas. Estados

En la tabla siguiente se describe el significado de cada uno de los estados.

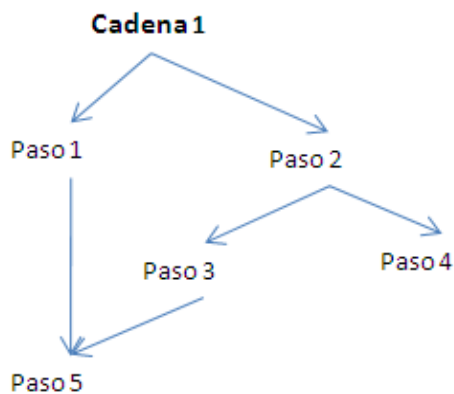
Estado	Descripción
SIN INICIAR	La Cadena al que pertenece el paso se está ejecutando, pero el paso no ha comenzado todavía.
PLANIFICADO	Una regla definida para el paso con una cláusula AFTER ha provocado que éste comience pero el tiempo designado por dicha cláusula no ha expirado todavía.
EN EJECUCIÓN	El paso se está ejecutando. Para un paso relacionado con un evento, el paso ha comenzado y está esperando que se produzca dicho evento.
PAUSADA	El paso ha sido interrumpido (pausado) Debe reactivarse antes de que los pasos que dependen de él puedan comenzar.
CORRECTO	El paso se ha completado correctamente. El código de error es 0.
INCORRECTO	El paso se ha completado con un error. El código de error es distinto de 0.
PARADO	El paso se ha parado mediante el procedimiento STOP_PROCEDURE.
ATASCADO	El paso hace referencia a una Cadena anidada que se encuentra en estado atascado.

Tabla 44: Manual Usuario. Gestión Cadenas. Estados

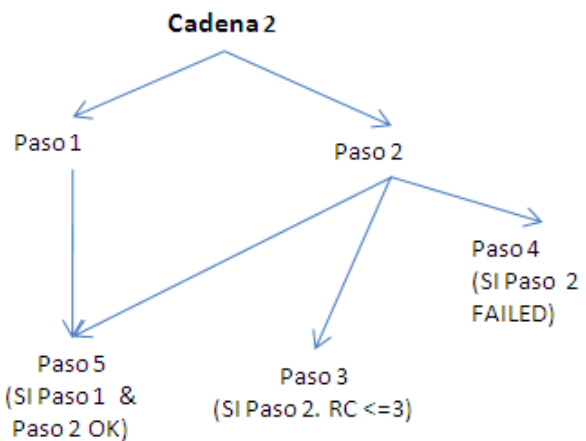
Definición de dependencias

Una vez definidos los pasos de una Cadena se deberán definir las dependencias y condiciones de ejecución de los mismos.

En los esquemas siguientes se representan dos ejemplos de las posibilidades que ofrecerá el nuevo sistema en cuanto a la definición de las dependencias entre los pasos de una cadena.



Ejemplo 1: Dependencias básicas de ejecución entre los pasos de una Cadena



Ejemplo 2: Dependencias de ejecución entre los pasos de una Cadena con condiciones basadas en resultados de pasos anteriores

Para que comience la ejecución de una Cadena, debe existir al menos un paso inicial. Se entiende como paso inicial, aquel que no depende de ningún otro paso de la Cadena y que su condición de ejecución se evalúa siempre a Verdadero.

Del mismo modo, para que una Cadena finalice su ejecución es necesario que contenga algún paso final. Se entiende como paso final, un paso para el que exista una dependencia que indique que e la Cadena finaliza cuando lo hace el paso.



Con el objetivo de asegurar la correcta definición de las Cadenas, junto con sus pasos y dependencias, se establecerá la siguiente casuística para el mantenimiento de dependencias:

- Para cada Cadena se definirá al menos un paso inicial.
- Para cada paso no inicial se establecerá una única condición de ejecución, que puede ser tan compleja como se precise utilizando las funciones lógicas AND, OR y NOT.
- Para cada Cadena se establecerá al menos una condición de finalización, que determinará el conjunto de validaciones que deben cumplirse para la finalización de dicha Cadena y el resultado de finalización del mismo. Para cada condición de finalización definida será necesario establecer:
 - Acción: Código de finalización o resultado con el que finalizará la Cadena si se cumple el criterio indicado en la condición.
 - Condición: validaciones sobre los estados (correcto, incorrecto, parado,...) de uno o varios de los pasos de la Cadena o sobre el código de error de los mismos.

A continuación se detallan algunos ejemplos sobre posibles condiciones de finalización para el ejemplo Cadena1 descrito con anterioridad.

Acción	Condición	Definición
END 0	PASO5 COMPLETED AND PASO5 SUCCEEDED	La cadena finaliza correctamente si el PASO5 finaliza correctamente.
END PASO5.ERROR_CODE	PASO5 COMPLETED AND PASO5 NOT SUCCEEDED	El proceso finaliza con error si el PASO5 finaliza de forma incorrecta. El Código de error es el mismo que el código de error que finaliza el PASO5
END -1	PASO4 COMPLETED	El proceso finalizará con error si el hilo de ejecución ha determinado que ejecute el PASO4

Diálogos y Pantallas

En esta opción se permitirá la navegación entre las distintas Cadenas de una aplicación. Para cada Cadena estarán disponibles las siguientes opciones:

- Resumen. En esta opción se podrá ver de forma conjunta los datos específicos de la Cadena, los pasos que los componen y para cada paso, sus predecesores y sucesores. Se trata de una opción de sólo consulta.
- Datos Generales. Atributos específicos de la Cadena.
- Mantenimiento de Pasos y Dependencias. Pasos que componen la Cadena y dependencias de ejecución entre los mismos.
- Tareas. Opción de consulta para visualizar el conjunto de Tareas en los que se encuentra referenciado la Cadena.

Resumen Cadena

La opción *Resumen* de un proceso será accesible desde la opción *Gestión de Procesos* del panel general de la aplicación.

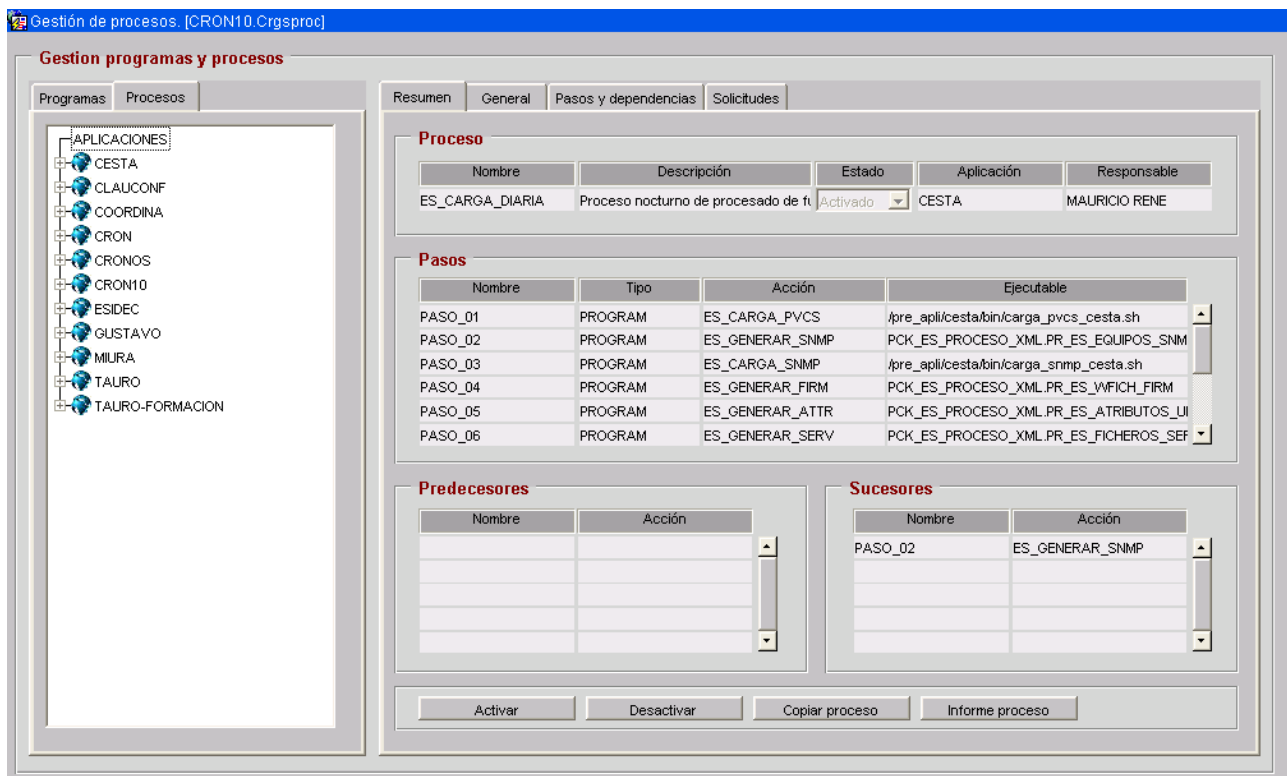


Ilustración 71: Manual Usuario. Gestión Cadenas. Resumen.

En esta opción se muestran para la Cadena actual el conjunto de pasos que lo componen, y para cada paso el conjunto de pasos predecesores y sucesores.



Activación de una Cadena

La activación de una Cadena permite que el mismo sea referenciado en una Tarea. Para activar una Cadena deben cumplirse las siguientes validaciones:

- Debe existir al menos un paso definido para la Cadena.
- Debe existir al menos una regla para alguno de los pasos de la Cadena que se hay definido como paso inicial.
- Debe existir una condición de finalización para la Cadena.

Des-activación de una Cadena

Para desactivar una Cadena deben cumplirse las siguientes validaciones:

- No debe existir ninguna Tarea activa en la que la Cadena se encuentre referenciado.

Copiar Cadena

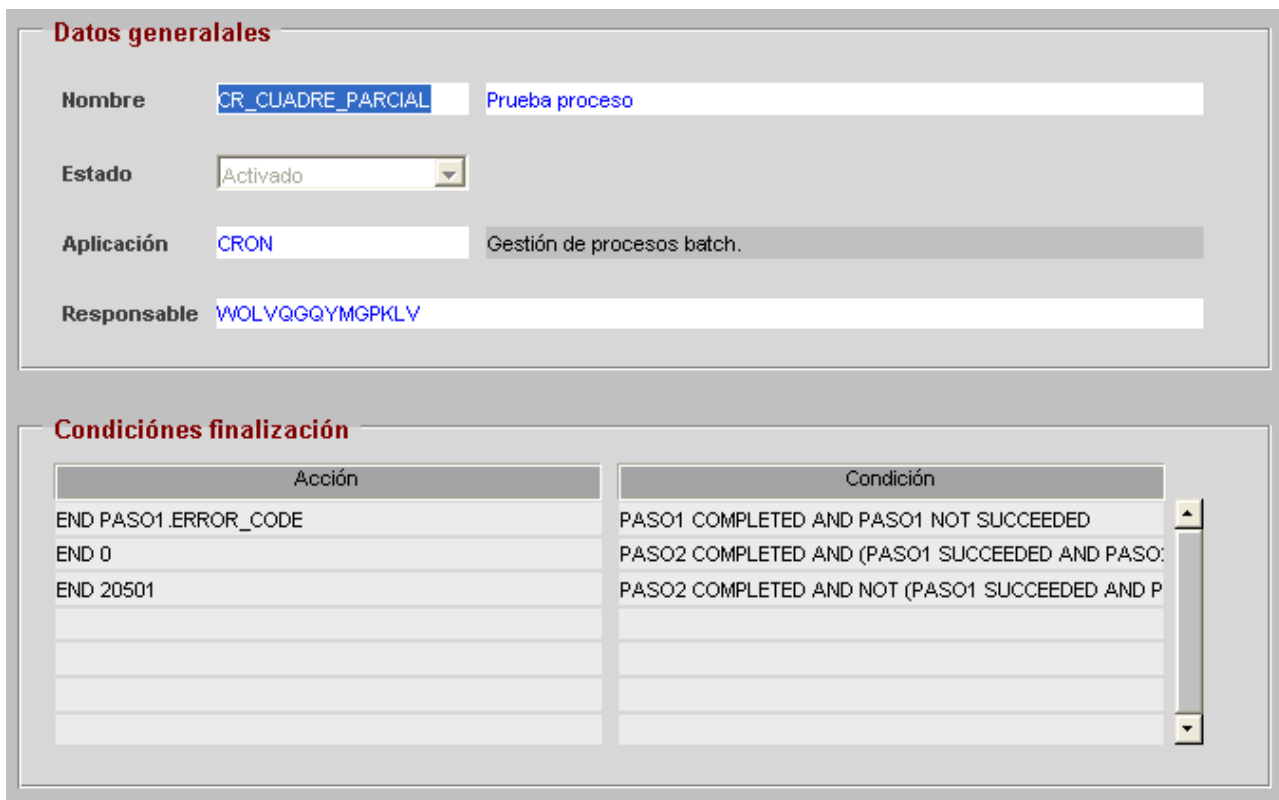
Esta funcionalidad permite copiar todos los datos de una Cadena a otra nueva.

Informe de Cadena

Esta funcionalidad genera un informe con la estructura jerárquica de pasos de la Cadena y el conjunto de dependencias entre ellos.

Mantenimiento de Cadenas (Datos Generales)

El *mantenimiento de Cadenas* será accesible desde la opción *Gestión de Procesos* del panel general de la aplicación en la pestaña *General*.



Datos generales

Nombre: Prueba proceso

Estado:

Aplicación: Gestión de procesos batch.

Responsable:

Condiciones finalización

Acción	Condición
END PASO1.ERROR_CODE	PASO1 COMPLETED AND PASO1 NOT SUCCEEDED
END 0	PASO2 COMPLETED AND (PASO1 SUCCEEDED AND PASO1 NOT SUCCEEDED)
END 20501	PASO2 COMPLETED AND NOT (PASO1 SUCCEEDED AND PASO1 NOT SUCCEEDED)

Ilustración 72: Manual Usuario. Gestión Cadenas. Datos Generales

A continuación se describen los campos asociados a una Cadena que serán informados en esta opción junto con sus condiciones de obligatoriedad y editabilidad:

Datos Generales			
Campo	Descripción / Comentarios	Oblig.	Edit.
Aplicación	Aplicación al que pertenece la Cadena	S	S
Nombre	Identificación de la Cadena	S	N
Descripción	Descripción de la Cadena	N	S
Estado	Estado de la Cadena: Activa o Desactiva.	S	N
Responsable	Usuario encargado del Mantenimiento de la Cadena.	S	S
Condiciones Finalización	Conjunto de reglas que se deben cumplir para considerar que la Cadena se ha finalizado.	S	S

Tabla 45: Manual Usuario. Gestión Cadenas. Datos Generales

Alta Cadena

Para realizar el alta de una Cadena deben cumplirse las siguientes validaciones:

- Deben estar informados todos los campos obligatorios
- No debe existir ningún objeto en SCHEDULER con el mismo nombre.
- Debe existir la aplicación a la que referencia.
- Debe existir el responsable.

Modificación Cadena

Se permitirá la modificación de los siguientes datos asociados a una Cadena:

- Descripción, Aplicación, Responsable
- Condiciones de Fin

Si se modifican las condiciones de finalización se validará que:

- Exista al menos una condición de finalización

Baja Cadena

Para realizar la baja de una Cadena deben cumplirse las siguientes validaciones:

- La Cadena no puede estar referenciado en ninguna Tarea
- La Cadena no podrá estar referenciado en otra Cadena como paso del mismo

Establecer condición

Esta opción mostrará una ventana para facilitar la introducción de las condiciones de finalización de la Cadena.

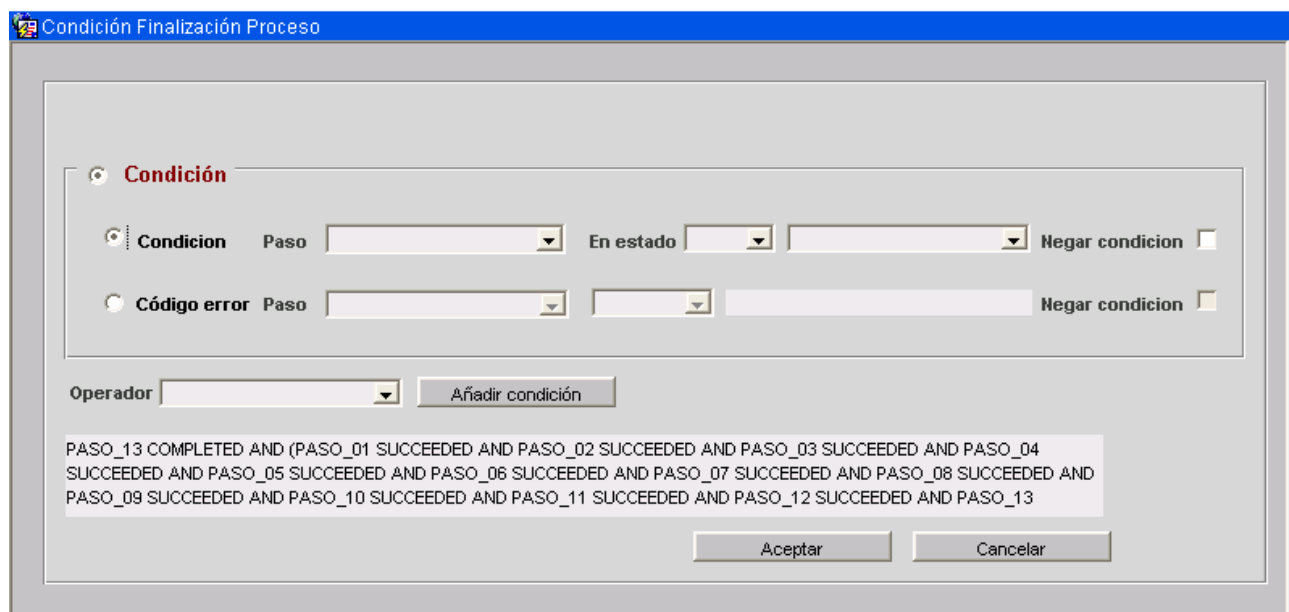
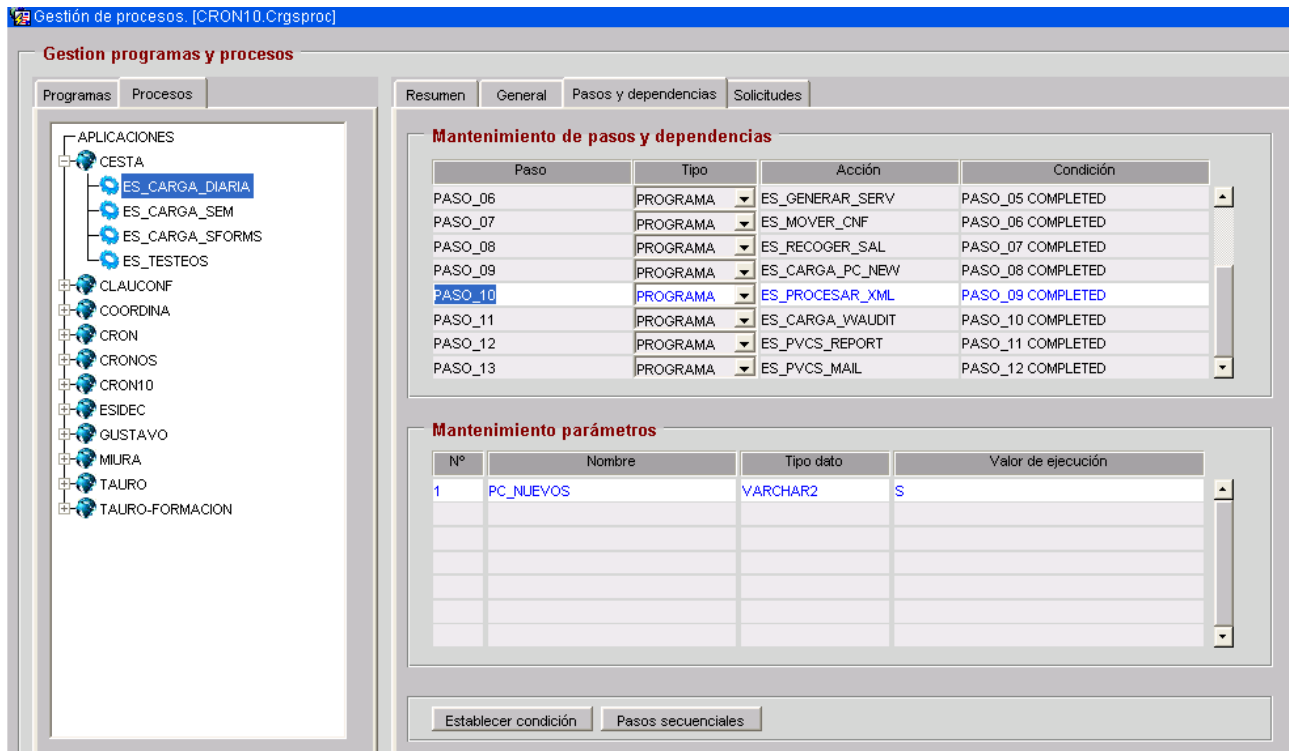


Ilustración 73: Manual Usuario. Gestión Cadenas. Condición Finalización

Mantenimiento de Pasos y Dependencias

El mantenimiento de los pasos asociados a una Cadena se realizará desde la pestaña Pasos/Dependencias mostrada en la opción de Gestión de Procesos.

En esta opción se mostrará para la Cadena actual una lista con todos los pasos que contiene la Cadena. Para cada paso que referencie a un programa se mostrará también la lista de parámetros asociados a dicho programa con los valores específicos con los que se ejecutará dicho programa en la Cadena actual.



The screenshot shows a web application interface for process management. On the left, there is a tree view of applications under 'APLICACIONES', with 'CESTA' expanded to show steps like 'ES_CARGA_DIARIA', 'ES_CARGA_SEM', 'ES_CARGA_SFORMS', and 'ES_TESTEOS'. The main area has tabs for 'Resumen', 'General', 'Pasos y dependencias', and 'Solicitudes'. The 'Pasos y dependencias' tab is active, displaying a table of steps and their dependencies. Below it, the 'Mantenimiento parámetros' section shows a table for step parameters.

Paso	Tipo	Acción	Condición
PASO_06	PROGRAMA	ES_GENERAR_SERV	PASO_05 COMPLETED
PASO_07	PROGRAMA	ES_MOVER_CNF	PASO_06 COMPLETED
PASO_08	PROGRAMA	ES_RECOGER_SAL	PASO_07 COMPLETED
PASO_09	PROGRAMA	ES_CARGA_PC_NEW	PASO_08 COMPLETED
PASO_10	PROGRAMA	ES_PROCESAR_XML	PASO_09 COMPLETED
PASO_11	PROGRAMA	ES_CARGA_WAUDIT	PASO_10 COMPLETED
PASO_12	PROGRAMA	ES_PVCS_REPORT	PASO_11 COMPLETED
PASO_13	PROGRAMA	ES_PVCS_MAIL	PASO_12 COMPLETED

Nº	Nombre	Tipo dato	Valor de ejecución
1	PC_NUEVOS	VARCHAR2	S

Ilustración 74: Manual Usuario. Gestión Cadenas. Pasos y Dependencias

A continuación se describen los campos asociados a un paso que serán informados en esta opción junto con sus condiciones de obligatoriedad y editabilidad

Pasos y Dependencias			
Campo	Descripción / Comentarios	Oblig.	Edit.
Nombre	Nombre del paso.	S	N
Tipo	Tipo de acción: Programa, Evento o Cadena.	S	S
Acción	Dependiendo del Tipo. La acción hará referencia al objeto que debe ejecutarse. Dicho objeto debe existir previamente en SCHEDULER:	N	S
Condición	Condición ejecución del paso.	S	N

Tabla 46: Manual Usuario. Gestión Cadenas. Pasos y Dependencias

En la siguiente tabla se describen los campos asociados a los parámetros de un paso que referencia a un programa que serán informados en esta opción, junto con sus condiciones de obligatoriedad y editabilidad.

Parámetros Paso			
Campo	Descripción / Comentarios	Oblig.	Edit.
Posición	Posición del paso.	S	N
Nombre	Nombre del parámetro.	S	N
Tipo	Tipo de parámetro: Numérico, carácter o Fecha.	S	N
Valor	Valor que se asocia al parámetro para ejecutar el paso.	N	S

Tabla 47: Manual Usuario. Gestión Cadenas. Parámetros Pasos

Alta de Paso

El alta de un paso desencadenará las siguientes acciones:

- Alta del paso y de las condiciones de ejecución del mismo en el sistema.
- Si el paso referencia a un programa con parámetros, alta de los parámetros de dicho programa con sus valores de ejecución para el proceso actual.

Modificación de Paso

La modificación de un paso desencadenará las siguientes acciones:

- Modificación del paso y de las condiciones de ejecución del mismo en las estructuras de SCHEDULER.
- Si el paso referencia a un programa con parámetros, modificación de los valores de ejecución de dichos parámetros.

Baja de Paso

La baja de un paso desencadenará las siguientes acciones:

- Si el paso hace referencia a un programa:
 - Baja de los parámetros si los tuviera.
- Baja del paso.
- Baja de las condiciones de ejecución del mismo.

Crear Pasos Secuenciales

Esta opción mostrará una ventana al usuario para que indique el número de pasos que desea crear. Una vez introducido este dato se mostrarán en pantalla el número de pasos indicado, estableciendo para los mismos un orden secuencial de ejecución, permitiendo al usuario definir el resto de datos necesarios para cada paso (tipo, programa/evento y parámetros si procede)

Establecer Condición

Esta opción mostrará una ventana para facilitar la introducción de las condiciones de ejecución del paso seleccionado.

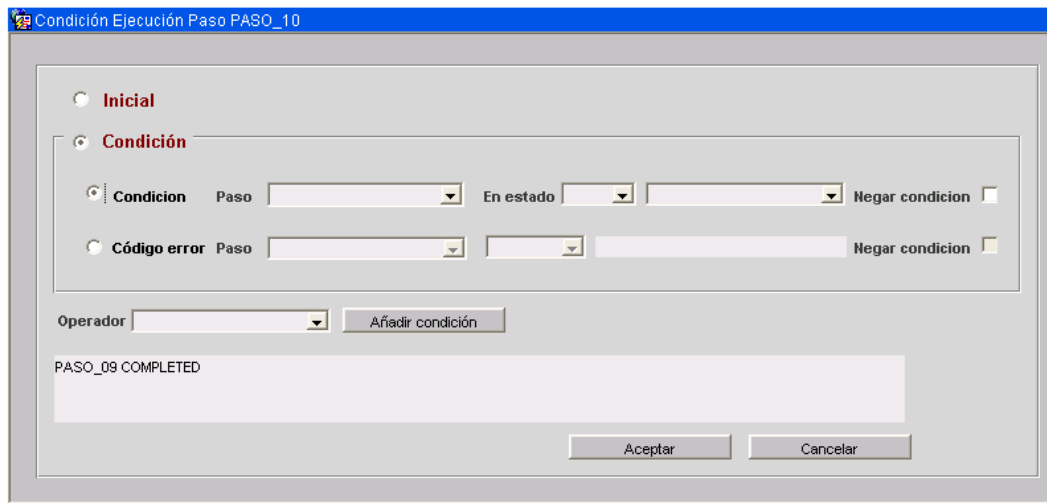


Ilustración 75: Manual Usuario. Gestión Cadenas. Condición Ejecución

Consulta de Tareas

La *Consulta de Tareas* en las que se encuentra referenciado la Cadena actual, se realizará desde la pestaña *Solicitudes* mostrada en la opción de *Gestión de Procesos*.

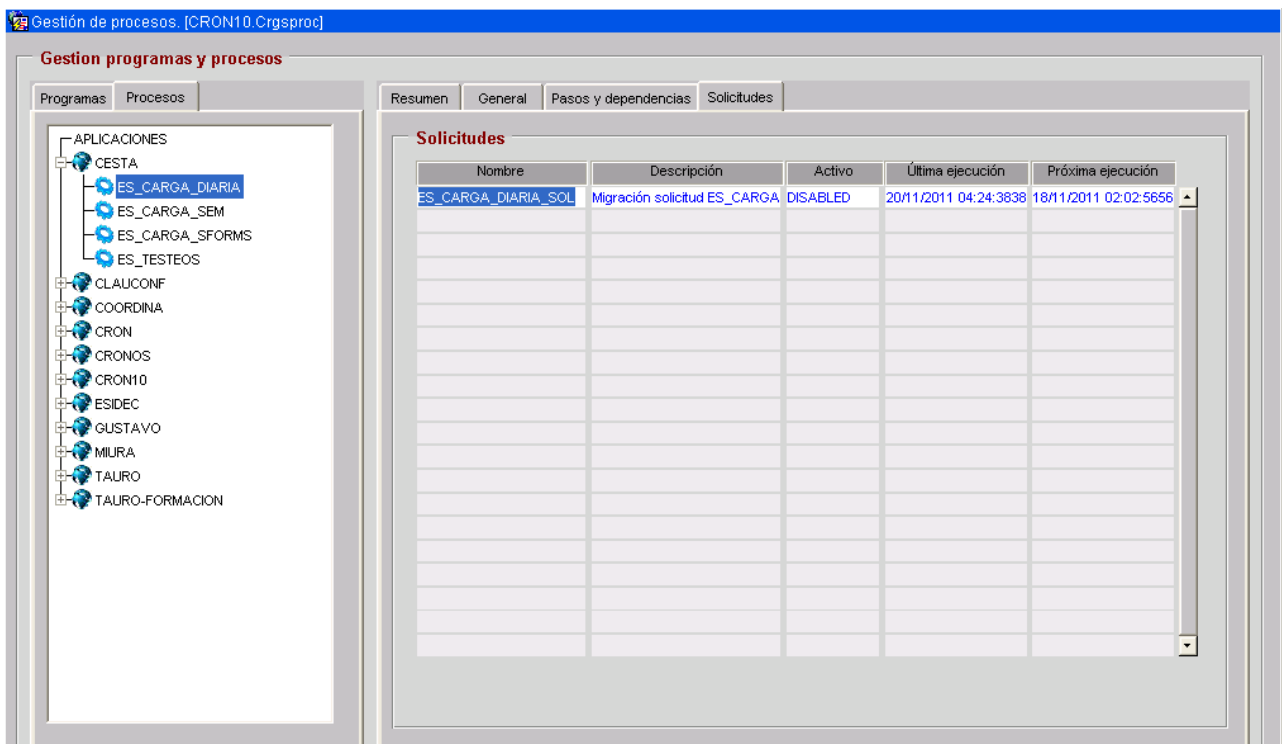


Ilustración 76: Manual Usuario. Gestión Cadenas. Consulta Tareas



Gestión de Tareas

Definición de Tareas

Una Tarea es una combinación de:

- Una Cadena (conjunto de programas con el valor de sus argumentos si procede) o un Programa
- Una planificación para la ejecución de los mismos.

Las Tareas son los objetos a través de los cuales, el nuevo sistema, realiza la ejecución de las Cadenas o Programas requeridos por cada aplicación. Al finalizar la ejecución de una Tarea se generará un evento que será utilizado por el Gestor de Resultados para registrar los resultados de la ejecución y activar las notificaciones asociadas a la misma. De esta forma se permitirá la suscripción de usuarios a los distintos resultados de una Tarea y a los ficheros que ésta genera.

Planificación de Tareas

La planificación de una Tarea determinará el momento en que la Tarea se planifica para su ejecución. Existen distintos tipos de planificaciones:

- Ejecución periódica: Planificación de una Tarea para que su ejecución se repita de forma periódica con una frecuencia fija (diariamente, semanalmente, mensualmente,...). En este tipo de planificaciones además de establecer dicha frecuencia se puede definir una fecha de inicio para determinar la primera ejecución y una fecha de fin para determinar la última.
- Asociada a un evento: Planificación de una Tarea para que su ejecución se condicione a la ocurrencia de un evento externo, existente en el catálogo de eventos del sistema.
- Ejecución inmediata: Planificación de una Tarea para que su ejecución se realice lo antes posible.

Estados de una Tarea

En una Tarea existen dos campos para determinar la situación en la que se encuentra:

- Indicador Activo: Flag o indicador que determina si una Tarea se encuentra habilitada. Por defecto las Tareas se crean con este indicador deshabilitado, y es necesario activarlo para que el planificador de SCHEDULER planifique la siguiente ejecución de una Tarea.
- Estado: El estado de una Tarea estará determinado por su situación (activa, desactiva), por el número de ejecuciones previstas y los resultados de estas ejecuciones.

A continuación se detallan los distintos estados en los que puede encontrarse una Tarea:

Estado	Descripción	Activa
DESHABILITADA	Tarea deshabilitada	NO
PLANIFICADA	Tarea planificada para ejecutarse	SI
EN EJECUCION	Tarea en ejecución	SI
CADENA_ATASCADA	Tarea asociada a una Cadena en estado Atascado. Una cadena llega a este estado si no hay pasos ejecutándose o planificados. Se precisa intervención manual.	SI / NO
INTERRUMPIDA	Tarea ejecutada de forma errónea y se ha superado el número de intentos de ejecución con error establecido.	NO
PARADA	Tarea parada cuando se estaba ejecutando.	NO



INCORRECTA	La Tarea ha finalizado con errores.	NO
CORRECTA	La Tarea ha finalizado correctamente.	NO
FINALIZADA	La Tarea ha finalizado todas las ejecuciones previstas.	NO

Tal y como queda reflejado en esta tabla, el estado en el que queda una Tarea tras una ejecución depende del número de ejecuciones previstas en la misma:

- Para aquellas Tareas planificadas para una única ejecución, al finalizar esta ejecución su estado se corresponderá con el estado de dicha ejecución (parada, incorrecta o correcta).
- En caso contrario, cuando están previstas más de una ejecución de la misma Tarea, al finalizar cada ejecución la Tarea volverá al estado planificada si existen más ejecuciones previstas o al estado finalizada si se trata de la última ejecución. En ambos casos, se podrán consultar los resultados de cada una de las ejecuciones de una Tarea.

La correspondencia entre resultados de ejecución y estados en los que se encuentra una Tarea, tras la última ejecución es la siguiente:

Resultado Ejecución	Estado Tarea
CORRECTA	CORRECTA PLANIFICADA FINALIZADA
INCORRECTA	INCORRECTA PLANIFICADA FINALIZADA INTERRUMPIDA
PARADA	PARADA PLANIFICADA FINALIZADA
ATASCADA	ATASCADA

Diálogos y Pantallas

La gestión de Tareas se realizará en las siguientes opciones:

- **Gestión de Tareas**
En esta opción se permitirá administrar las Tareas existentes en el sistema.
- **Mantenimiento de Tareas**
En esta opción se permitirá el alta, modificación y eliminación de Tareas y de los datos asociados a las mismas.
- **Mantenimiento de Parámetros de una Tarea**
En esta opción se permitirá el alta, modificación y eliminación de los valores de ejecución de los parámetros de una Tarea, si ésta referencia a un programa.
- **Monitorizar Ejecución**
En esta opción se podrá visualizar el estado de cada uno de los pasos de una Tarea que referencia a un Cadena y se encuentre en ejecución.
- **Mantenimiento de Suscripciones**
Esta opción permite definir las suscripciones de un usuario a una tarea

Gestión Tareas

En esta opción se puede visualizar el Catálogo de Tareas existentes en el sistema. Para acceder a las mismas existirán dos opciones:

- Navegar entre las distintas Tareas de una aplicación, asumiendo que la aplicación de una Tarea será la aplicación a la que pertenece la Cadena o Programa referenciado en la misma.
- Realizando una búsqueda por cualquiera de los criterios de búsqueda disponibles en la pantalla.

A continuación se describen los campos asociados a una Tarea que se visualizarán en esta opción junto con sus condiciones de obligatoriedad y editabilidad.

Catálogo Tareas			
Campo	Descripción / Comentarios	Oblig.	Edit.
Tarea	Identificador de Tarea	S	N
Activado	Activa / Desactiva	S	N
Estado	Estado de la Tarea	S	N
Última ejecución	Fecha última ejecución de la Tarea	S	N
Próxima Ejecución	Próxima fecha que se ejecuta la Tarea	N	N
Acción	Cadena o Programa que ejecuta la Tarea	S	N
Aplicación	Aplicación a la que pertenece	S	N
Responsable	Usuario responsable de la Tarea	S	N

Tabla 48: Manual Usuario. Gestión Tareas. Catálogo Tareas

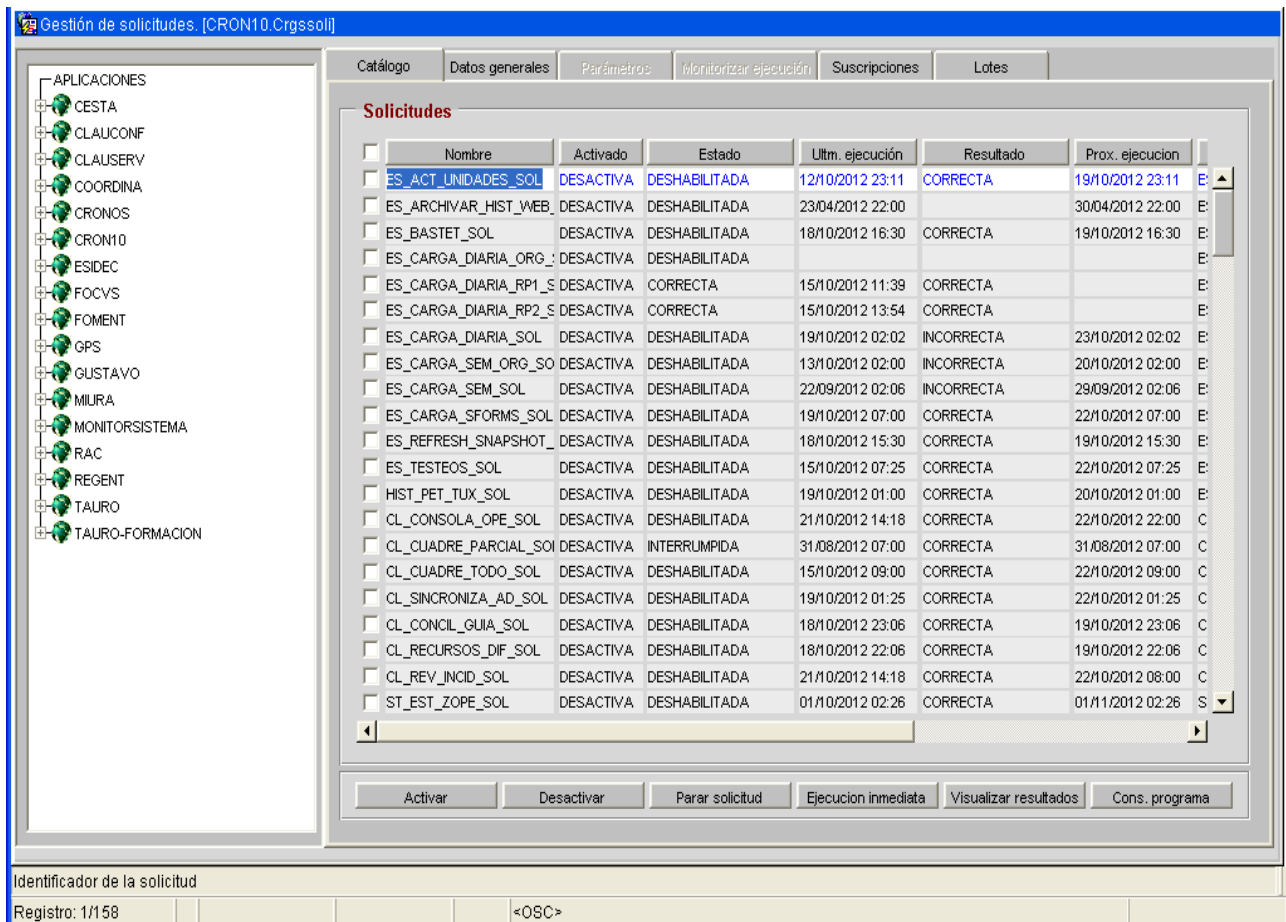


Ilustración 77: Manual Usuarios. Gestión Tareas. Catálogo

Activación de una Tarea

Permite activar una Tarea, al realizar esta acción SCHEDULER planificará la próxima ejecución de la Tarea en función de la planificación definida para la misma.

Des-activación de una Tarea

Al desactivar una Tarea, se desplanificará su siguiente ejecución.

Para una Tarea

Esta opción permite finalizar la ejecución de una Tarea forzando un resultado de ejecución de Parado.

Para parar una Tarea deben cumplirse las siguientes validaciones:

- La tarea debe estar en ejecución

Al parar una Tarea, se desencadenarán las siguientes acciones:

- Se finaliza la ejecución, realizando ROLLBACK de la transacción actual.
- Si la Tarea sólo tenía prevista una ejecución, su estado pasará a parada.



- Si la Tarea tenía planificadas más de una ejecución su estado pasará a planificada.

Ejecución Inmediata de una Tarea

Esta opción permitirá seleccionar una o varias Tareas con distintas planificaciones y crear una petición para ejecutarlas inmediatamente. Las Tareas seleccionadas comenzarán a ejecutarse tan pronto como el planificador de SCHEDULER pueda asignarles los recursos necesarios.

Consultar Programa/Cadena

El botón Consultar Programa/Proceso, desencadenará las siguientes acciones:

- En función del tipo de Tarea seleccionado navegará a la pantalla de Gestión de Cadenas o Programas, mostrando la información relativa a la Cadena o Programa referenciado por la Tarea.

Mantenimiento Tareas

El *Mantenimiento de Tareas* será accesible desde la opción *Gestión de Tareas* del panel general de la aplicación en la pestaña Datos Generales.

The screenshot shows a web application window titled 'Gestión de solicitudes. [CRON10.Crgssoll]'. The interface includes a navigation tree on the left with various application categories like 'CESTA', 'CLAUCONF', 'CLAUSERV', etc. The main area is divided into tabs: 'Catálogo', 'Datos generales', 'Parámetros', 'Monitorizar ejecución', 'Suscripciones', and 'Lotes'. The 'Datos generales' tab is active, displaying the details for a task named 'ES_ACT_UNIDADES_SOL' with the subtitle 'Migración solicitud ES_ACT_UNIDADES_SOL'. The task configuration includes fields for 'Tipo' (PROGRAMA), 'Acción' (ES_ACT_UNIDADES), 'Clase' (DEFAULT_JOB_CLASS), 'Estado' (Desactivado), 'Prioridad' (3), and 'Nro. maximo errores' (5). It also shows 'Historización' (VIGENCIA 1 MES) and 'Genera incidencia' (NO_GENERAR). The 'Tipo ejecución' section has radio buttons for 'Inmediato', 'Evento', 'Predefinida', and 'Específica', with 'Específica' selected. Below this, there are input fields for 'Fecha inicio', 'Fecha fin', and 'Intervalo repetición' (FREQ=WEEKLY;INTERVAL=1;BYHOUR=23;BYMINUTE=11;BYDAY=FR;). The 'Proximas ejecuciones' section shows a list of upcoming runs. At the bottom, there are statistics for 'Ejecuciones': Nro. ejecuciones (6), Nro. ejecuciones correctas (6), Nro. ejecuciones incorrectas (0), and Nro. ejecuciones paradas (0). Buttons for 'Establecer planificación' and 'Proximas ejecuciones' are visible. The status bar at the bottom indicates 'Identificación de la solicitud en el sistema' and 'Registro: 1/1'.

Ilustración 78: Manual Usuario. Gestión Tareas. Datos Generales

A continuación se describen los campos asociados a una Tarea que serán informados en esta opción junto con sus condiciones de obligatoriedad y editabilidad.



Datos Generales			
Campo	Descripción / Comentarios	Oblig.	Edit.
Nombre	Identificador de Tarea	S	N
Descripción	Descripción de la tarea	S	S
Tipo	Tipo de Objeto referenciado por la Tarea (Programa / Cadena)	S	S
Acción	Nombre del Programa / Cadena	S	S
Clase	Clase de Tarea asociada.	S	S
Estado	Activo / Desactivo.	S	N
Prioridad	Prioridad de la Tarea respecto a otras tareas pertenecientes a la misma Clase	S	S
Nº Máximo Errores	Nº Máximo de errores consecutivos que puede tener la tarea cuando se ejecut.,	S	S
Historización	Tipo de Historización aplicado a la Tarea.	S	S
Genera incidencia	Indicador de generación de incidencia.	S	S
Tipo ejecución	Se permite cuatros tipos de ejecución: <ul style="list-style-type: none">• Inmediata• Evento: Indicar el Evento Externo• Predefinida: Indicar Planificación• Específica: indicar:<ul style="list-style-type: none">• Fecha Inicio• Fecha Fin• Intervalo de Repetición	S	S
Ejecuciones	Determina el Número de ejecuciones que ha tenido la Tarea: <ul style="list-style-type: none">• Total Ejecuciones• Ejecuciones Correctas• Ejecuciones Incorrectas• Ejecuciones Paradas	N	N

Tabla 49: Manual Usuario. Gestión Tareas. Datos Generales.

Alta Tarea

Para realizar el alta de una tarea deben cumplirse las siguientes validaciones:

- Si la Tarea es de tipo Programa, debe existir el Programa referenciado.
- Si la tarea es de tipo Cadena, debe existir la Cadena referenciado.
- Debe existir la Clase de Tarea referenciada.
- No debe existir ningún objeto SCHEDULER con el mismo nombre.
- Debe estar informada la planificación de la Tarea con alguna de las siguientes opciones excluyentes entre sí:
 - Inmediata
 - Evento Iniciador
 - Planificación predefinida



- Planificación específica

Modificación Tarea

Para realizar la modificación de una Tarea deben cumplirse las siguientes validaciones:

- Deben estar informados todos los campos obligatorios.
- Si la Tarea es de tipo Programa, debe existir el Programa referenciado.
- Si la Tarea es de tipo Cadena, debe existir la Cadena referenciado.
- Debe existir la Clase de Tarea referenciada.
- Debe estar informada la planificación de la Tarea con alguna de las siguientes opciones excluyentes entre sí:
 - Inmediata
 - Evento Iniciador
 - Planificación predefinida
 - Planificación específica

Baja Tarea

Para realizar la baja de una Tarea deben cumplirse las siguientes validaciones:

- La Tarea no debe estar ejecutándose

La baja de una Tarea desencadenará las siguientes acciones:

- Baja de la Tarea de las estructuras de SCHEDULER.
- Baja de los parámetros
- Baja de las suscripciones

Establecer Planificación

Esta opción muestra el Formulario de *Detalle de Planificación* que permite la introducción de la planificación de la Tarea ofreciendo al usuario casillas de selección y un formato gráfico más amigable que la sintaxis de calendario de Scheduler.

Consultar Próximas Ejecuciones

Esta funcionalidad permitirá consultar las próximas fechas en las que una Tarea se planificará para su ejecución según la planificación definida.

Para realizar esta consulta deben cumplirse las siguientes validaciones:

- Debe estar seleccionada una solicitud
- La solicitud debe estar planificada mediante una planificación predefinida o específica. No debe estar asociada a un evento, ni ser inmediata
- Debe indicarse el número de próximas ejecuciones que se desea mostrar

La consulta de próximas ejecuciones desencadenará las siguientes acciones:

- Mostrará una lista de fechas con las próximas ejecuciones previstas de la solicitud. El número de ejecuciones mostrado coincidirá con el indicado en la pantalla.

Parámetros Tareas

El mantenimiento de los parámetros de ejecución de una Tarea será accesible desde la opción Gestión de Solicitudes del panel general de la aplicación en la pestaña Parámetros.

Esta funcionalidad permitirá modificar los valores de ejecución de los parámetros de una Tarea que referencia a un programa.

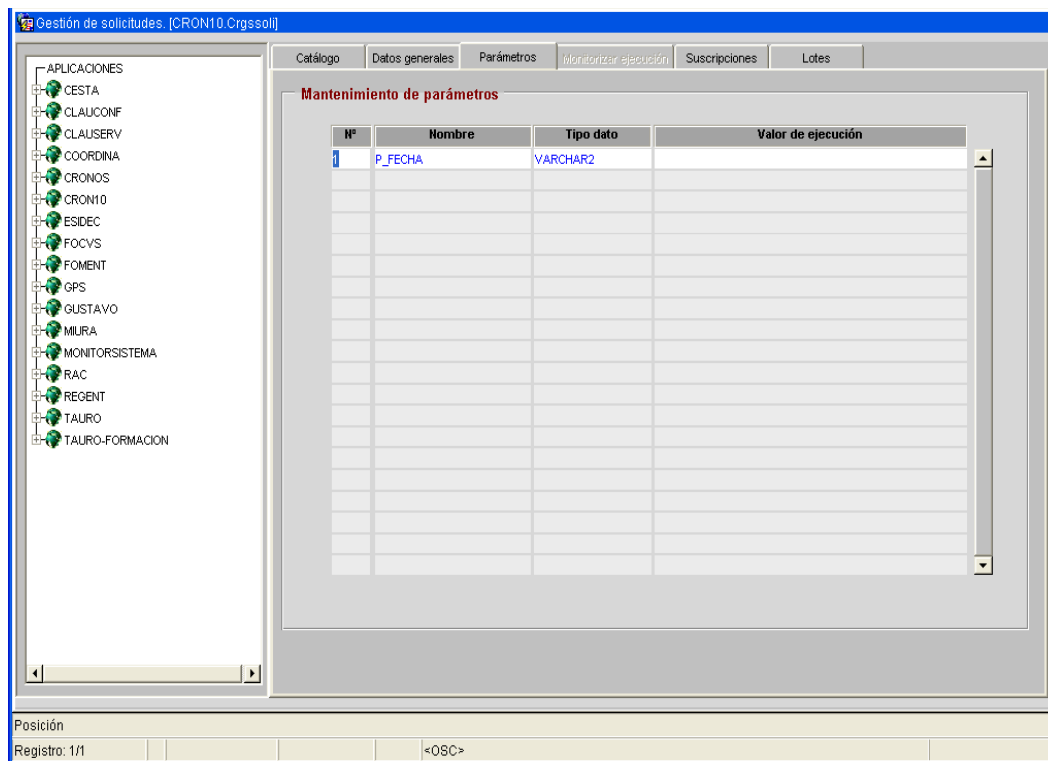


Ilustración 79: Manual Usuario. Gestión Tareas. Parámetros

A continuación se describen los campos asociados a los parámetros de una Tarea que serán informados en esta opción junto con sus condiciones de obligatoriedad y editabilidad.

Parámetros Tareas			
Campo	Descripción / Comentarios	Oblig.	Edit.
Posición	Posición del parámetro	S	N
Nombre	Nombre parámetro	S	N
Tipo	Tipo Parámetro: numérico, Carácter o Fecha.	S	N
Valor	Valor asociado al parametro	N	S

Tabla 50: Manual Usuario. Gestión Tareas. Parámetros

Monitorizar Ejecución

Esta opción será accesible desde la opción *Gestión de Solicitudes* del panel general de la aplicación en la pestaña *Monitorizar Ejecución*.

Esta funcionalidad permitirá visualizar la ejecución de una Tarea que referencia una Cadena e interactuar con ella, bien parando su ejecución bien modificando el estado de alguno de los pasos.

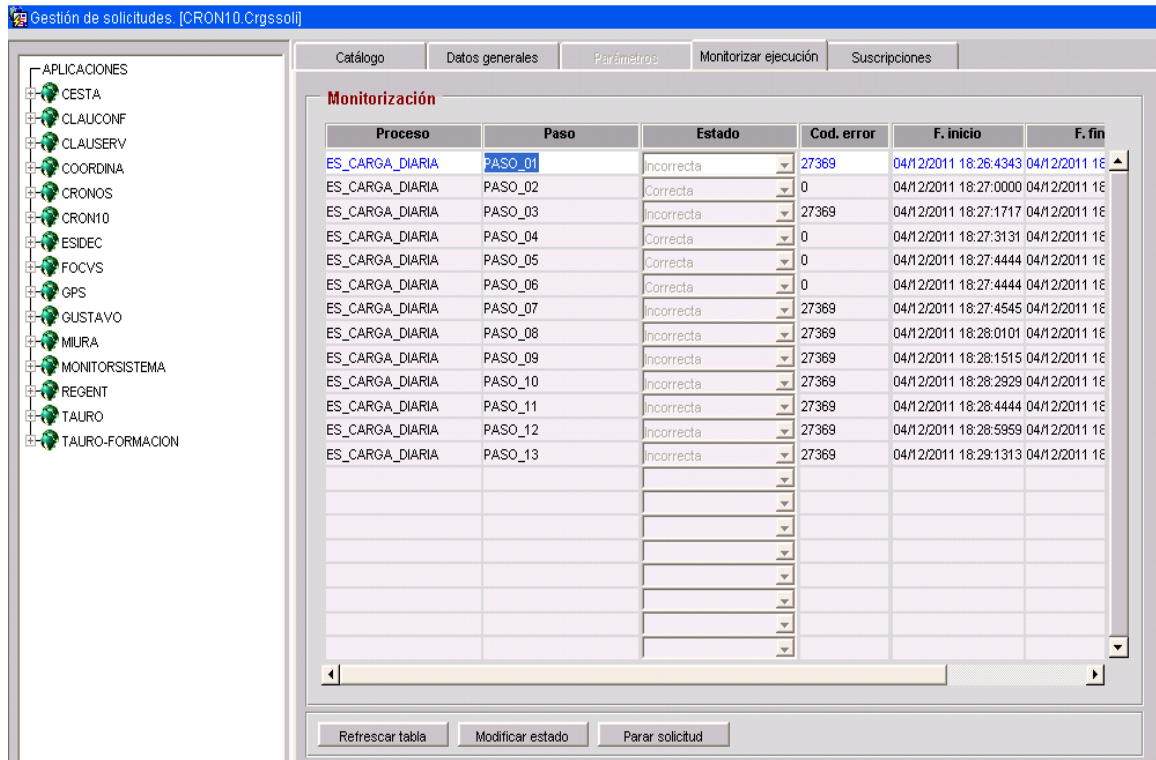


Ilustración 80: Manual Usuario. Gestión Tareas. Monitorizar Ejecución.

A continuación se describen los campos asociados a los parámetros de una Tarea que serán informados en esta opción junto con sus condiciones de obligatoriedad y editabilidad.

Monitorizar Ejecución			
Campo	Descripción / Comentarios	Oblig.	Edit.
Cadena	Identificador de Cadena	N	N
Paso	Nombre del Paso	N	N
Estado	Estado actual del Paso	N	N
Código Error	Código Error en que finaliza el paso	N	N
Fecha Inicio	Fecha que ha empezado la ejecución del paso	N	N
Fecha Fin	Fecha que ha finalizado la ejecución del paso.	N	N

Tabla 51: Manual Usuario. Gestión Tareas. Monitorizar Ejecución.



Refrescar

Esta opción permite refrescar la información sobre la ejecución de la Tarea actual.

Parar Tarea

Esta opción permite finalizar la ejecución de la Tarea forzando un resultado de ejecución de Parado. Al parar la Tarea, se desencadenarán las siguientes acciones:

- Se finaliza la ejecución, realizando ROLLBACK de la transacción actual.
- Si la Tarea sólo tenía prevista una ejecución, su estado pasará a parada.

Modificar Estado Paso

Esta opción permite modificar el estado de un paso para alterar la ejecución de una Tarea. Puede ser útil para resolver situaciones en las que errores en la ejecución de uno de los pasos de la Cadena impida la ejecución de los pasos sucesores, y se requiera, tras una intervención manual para solucionar los errores surgidos, continuar con la ejecución de dichos paso.

Para realizar esta acción deben cumplirse las siguientes validaciones:

- El paso a modificar no debe estar en ejecución
- El paso final al que se quiere modificar el paso debe ser: Sin Iniciar, Correcto o Incorrecto. En este último caso deberá indicarse código de error.

Al realizar esta acción, se desencadenarán las siguientes acciones:

- Se modificará su estado al seleccionado por el usuario.
- Se continuará con la ejecución de la Cadena al que pertenezca el paso modificado según la definición de dependencias actual.

Suscripciones de una Tarea

Esta opción será accesible desde la opción *Gestión de Solicitudes* del panel general de la aplicación en la pestaña *Suscripciones*.

Correo	Móvil	Nombre	Apellido 1	Apellido 2	Condición	Error	Activar	Desact.
erey4617@gmail.com		ESTHER	FERNANDEZ	REY	Siempre		10/08/2011	
alonso_mau@gva.es		MAURICIO RENE	ALONSO	CANEIRO	Siempre		10/08/2011	
jalon_jos@gva.es		JOSE FRANCISC	JALON	SOLER	Siempre		10/08/2011	

Selección ficheros de la suscripción

Nombre del Fichero/Función	Condición
circonope.pdf	<input checked="" type="checkbox"/> Siempre
circonope.pdf.log	<input checked="" type="checkbox"/> Siempre
	<input type="checkbox"/> ▼
	<input type="checkbox"/> ▼
	<input type="checkbox"/> ▼
	<input type="checkbox"/> ▼
	<input type="checkbox"/> ▼
	<input type="checkbox"/> ▼
	<input type="checkbox"/> ▼

Ilustración 81: Manual Usuario. Gestión Tareas. Suscripciones.

Esta funcionalidad permitirá suscribir y de-suscribir a distintos usuarios a la Tarea actual. Se define el Concepto de Suscripción en el Apartado de *Gestión de Suscripciones*.



Gestión de Suscripciones

Definición de Suscripción

Una suscripción hace posible que un usuario reciba información del resultado de las Tareas que se han ejecutado así como de los resultados obtenidos en los mismos.

Una suscripción relaciona un usuario con una Tarea y define cuándo se quiere recibir información de la ejecución de la misma, así como qué información se quiere recibir.

La recepción de información podrá ser:

- Por correo electrónico. En la suscripción por correo electrónico se remitirá al usuario la información sobre la ejecución de la Tarea adjuntando los ficheros de resultado que se hayan definido.
- Por SMS. Se podrá enviar un SMS con la información sobre el resultado de la ejecución de la Tarea.

Diálogos y Pantallas

El mantenimiento de suscripciones será accesible:

- Desde el mantenimiento de Tareas, una vez seleccionada una Tarea, a través de la pestaña "Suscripciones". A través de esta opción se podrá suscribir a distintos usuarios a la Tarea actual.
- Desde la opción "Suscripciones" del panel general de la aplicación. A través de esta opción un usuario podrá suscribirse y/ eliminar suscripciones a distintas Tareas.

Suscripciones por Tarea

Catálogo	Datos generales	Parámetros	Monitorizar ejecución	Suscripciones				
Correo	Móvil	Nombre	Apellido 1	Apellido 2	Condición	Error	Activar	Desact.
erey4617@gmail.com		ESTHER	FERNANDEZ	REY	Siempre		10/08/2011	
alonso_mau@gva.es		MAURICIO RENE	ALONSO	CANEIRO	Siempre		10/08/2011	
jalon_jos@gva.es		JOSE FRANCISC	JALON	SOLER	Siempre		10/08/2011	

Selección ficheros de la suscripción

Nombre del Fichero/Función	Condición
circonope.pdf	Siempre
circonope.pdf.log	Siempre

Personalizar

Ilustración 82: Manual Usuarios. Gestor Suscriptores. Suscripciones por Tarea

Suscripciones por usuario

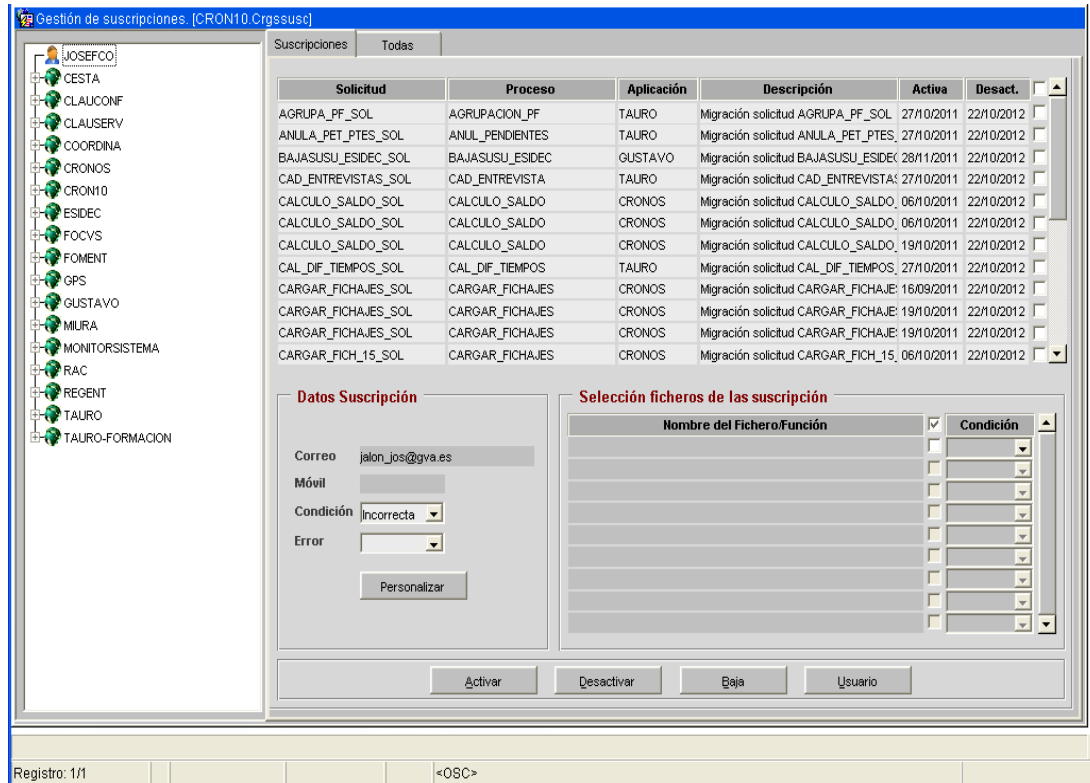


Ilustración 83: Manual Usuario. Gestor Suscripciones. Suscripciones por Usuario

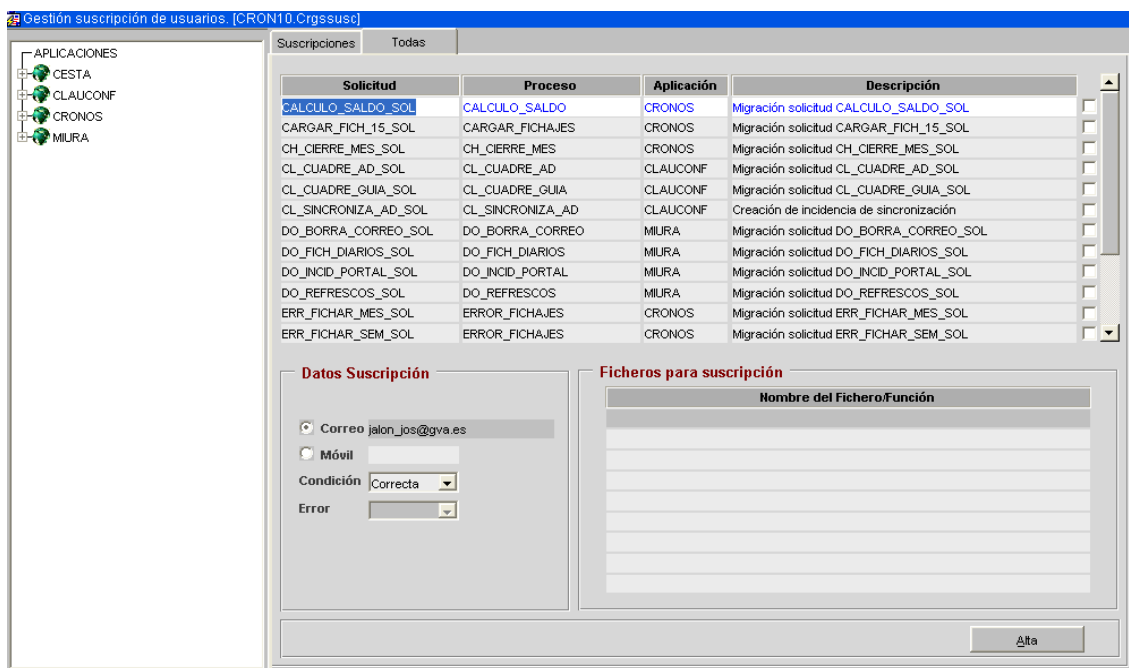


Ilustración 84: Manual Usuario. Gestor Suscripciones. Suscripciones por Usuario. Todas

A continuación se describen los campos asociados a una suscripción que serán informados en el mantenimiento de suscripciones, ya sea por usuario o por Tarea, junto con sus condiciones de obligatoriedad y editabilidad.

Datos Suscripción			
Campo	Descripción / Comentarios	Oblig.	Edit.
Tarea	Identificador de Tarea	S	S
Dirección Correo	Dirección de correo electrónico al que hay que enviar la notificación.	S	S
Móvil	Número de móvil al que hay que enviar SMS	S	S
Condición de envío	Situación en la que se quiere recibir la notificación, en función del resultado de la ejecución de la Tarea <ul style="list-style-type: none"> • Completo. Una vez finalizada • Correcto. Si ha finalizado con éxito • Incorrecto. Si ha finalizado con error • Parada. Si se ha detenido la ejecución de la Tarea • Atascada. Si la Tarea se encuentra en este estado 	S	S
Código de error	Especifica los códigos de error para los que se desea recibir la notificación, si la condición de envío especificada es <i>Incorrecta</i>	N	S
Fecha activación	Fecha a partir de la cual se empieza a enviar notificaciones.	S	S
Fecha de des-activación	Fecha a partir de la cual se de deja de enviar notificaciones.	N	S
Asunto Correo ¹²	Texto que se incluye en el Asunto del Correo	N	S
Cuerpo Correo	Texto que se incluye en el Cuerpo del Correo	N	S
Origen Correo	Dirección de Correo Electrónico que se especifica en el campo remitente.	N	S

Tabla 52: Manual Usuario. Gestor Suscripciones. Datos Suscripción

Para cada suscripción será necesario indicar a qué ficheros de los que genera la Tarea se desea realizar la suscripción. Para ello se mostrará una lista con los ficheros disponibles generados por el proceso ejecutado por la Tarea, para que el usuario escoja a qué ficheros desea suscribirse y en qué situación.

A continuación se describen los campos asociados a cada fichero de una suscripción un que serán informados en esta opción junto con sus condiciones de obligatoriedad y editabilidad.

Ficheros Suscripción			
Campo	Descripción / Comentarios	Oblig.	Edit.
Nombre Fichero / Función	Resultado al que se quiere suscribir.	S	N
Condición de Envío	Situación en la que se quiere recibir la notificación, <ul style="list-style-type: none"> • Completo. Una vez finalizada • Correcto. Si ha finalizado con éxito • Incorrecto. Si ha finalizado con error • Parada. Si se ha detenido la ejecución de la Tarea • Atascada. Si la Tarea se encuentra en este estado 	S	S

Tabla 53: Manual Usuario. Gestor Suscripciones. Ficheros Suscripción.

¹² Los Campos Asunto, Cuerpo y Origen están disponibles a través del botón *Personalizar*.



Alta Suscripción

Para realizar el alta de una suscripción deben cumplirse las siguientes validaciones:

- Deben estar informados todos los campos obligatorios.
- Debe existir la solicitud referenciada.
- Debe estar informada la dirección de correo electrónico o el número de móvil.
- Si el campo dirección de correo no está informado no se pueden seleccionar ficheros para la suscripción.
- Si el campo dirección de correo no está informado no se permitirá informar los campos Asunto, Texto y Origen del Correo.
- Sólo si la condición de envío es "Incorrecto" se podrá indicar el código de error.
- La condición de envío especificada para la recepción de ficheros anexados debe ser subconjunto de la condición de envío de la suscripción¹³

Modificación Suscripción

Para realizar la modificación de una suscripción deben cumplirse las siguientes validaciones:

- Deben estar informados todos los campos obligatorios.
- Debe existir la solicitud referenciada.
- Debe estar informada la dirección de correo electrónico o el número de móvil.
- Si el campo dirección de correo no está informado no se pueden seleccionar ficheros para la suscripción.
- Si el campo dirección de correo no está informado no se permitirá informar los campos Asunto, Texto y Origen del Correo.
- Sólo si la condición de envío es "Incorrecto" se podrá indicar el código de error.
- La condición de envío especificada para la recepción de ficheros anexados debe ser subconjunto de la condición de envío de la suscripción

Baja Suscripción

La baja de una suscripción desencadenará las siguientes acciones:

- Baja de suscripción y de ficheros asociados

Activación Suscripción

Para activar una suscripción deben cumplirse las siguientes validaciones:

- La suscripción debe estar desactiva.

La activación de una suscripción desencadenará las siguientes acciones:

¹³ No se permite la suscripción a un fichero en caso de error, si la condición de envío asociada a la suscripción es con estado Correcta.

Si que se permitirá la suscripción de ficheros en el caso de error cuando la condición de envío asociada a la suscripción es SIEMPRE.

- Modificación de la fecha de activación a la fecha indicada por el usuario

Des-activación Suscripción

Para desactivar una suscripción deben cumplirse las siguientes validaciones:

- La suscripción debe estar activa.

La des-activación de una suscripción desencadenará las siguientes acciones:

- Modificación de la fecha de des-activación a la fecha indicada por el usuario

Personalizar

Esta opción permite personalizar el correo electrónico que se recibirá en la suscripción.

Los campos parametrizables son: asunto, cuerpo y origen del correo.

Usuario

Solo los usuarios con perfil de Administración tienen accesible esta opción que permite consultar las suscripciones de otros usuarios.

Esta opción muestra la siguiente ventana de dialogo. Para agilizar la búsqueda se facilita una lista de valores con los distintos usuarios con suscripciones.

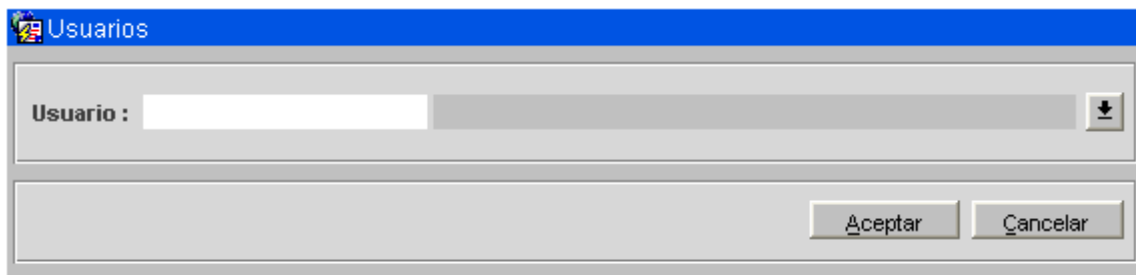


Ilustración 85: Manual Usuario. Gestión Suscripciones. Usuarios

Una vez seleccionado y aceptado el usuario, la ventana principal de Gestión de Suscripciones mostrará todas las Tareas en el que está suscrito el usuario que se ha seleccionado.

De este modo se permite a los usuarios con perfil de Administrador Activar y Desactivar suscripciones de un usuario distinto al conectado.

7 Conclusiones y trabajo futuro

En este proyecto se ha desarrollado una aplicación para el manejo del paquete **DBMS_SCHEDULER** de Oracle que permite definir las Tareas, que se ejecutan de forma diferida, a los *Administradores de Aplicaciones*, sin que estos deban encargarlas, como era habitual en nuestra administración, a los *Administradores de Sistemas*. El grave problema de esta forma de proceder es que quienes disponen del conocimiento para la definición de las Tareas diferidas son los *Administradores de Aplicaciones*, pero quien debían llevar al cabo dicha definición son los *Administradores de Sistemas*. Así, errores debidos a malentendidos son frecuentes y el mantenimiento muy complicado.

Esta aplicación, como se menciona, permite que sean los Administradores de Aplicación quienes definen las Tareas diferidas. Con ello el trabajo se agiliza porque se reducen errores debidos a mala comunicación. Evidentemente, todo esto supone un gran ahorro en horas de trabajo.

Tras la instalación de la aplicación en la *Administración Pública* de la *Generalitat Valenciana* se concluye que **el objetivo ha sido cumplido de forma satisfactoria**. No obstante, se detectan mejoras que pueden favorecer esta aplicación y ampliar sus funcionalidades:

- Modulo Gestión Resultados

Este Modulo permitiría la visualización del histórico de ejecuciones de las Tareas, permitiendo la posibilidad de descargar los ficheros Resultado de cada ejecución.

- Modulo Gestión de Lotes

Este Modulo permitiría agrupar las Tareas según interés del desarrollador, de esta forma, se pueden realizar las acciones comunes a un grupo de Tareas.

- Modulo Exportación / Importación

Este Modulo permitiría la creación de ficheros (Exportación) de Programas, Cadenas o Tareas , para poder ser creados (Importados) en otro servidor que tenga instalada la estructura de la aplicación.

Otras aplicaciones para el ámbito de **Oracle Database** consistiría en desarrollar interfaces gráficas, como las que se ha realizado en este proyecto, para crear aplicaciones basadas en otros paquetes de su suite, lo cual facilitaría, como se ha demostrado, el trabajo del desarrollador.





8 Bibliografía

- [1] Oracle Database, Administrator's Guide, 10g Release 2 (10.2)
- [2] Oracle Database, PL/SQL Packages and Types Reference, 10g Release 2 (10.2)
- [3] Oracle JDeveloper 10g for Forms & PL/SQL Developers : A Guide to Web Development with Oracle ADF (Oracle Press); Editorial McGraw-Hill Osborne Media; Edición: 1
- [4] Oracle Streams, Concepts and Administration, 10g Release 2 (10.2)
- [5] Oracle Streams, Advanced Queuing User's Guide and Reference, Release 10.1