Caravel OS/400 Framework

Visión general

BASE 100, S.A.
Santa María Magdalena, 10-12
28016 – Madrid
Tel.: 91 353 18 15
www.base100.com

BASE100

Índice

1. Introducción

El proceso de conversión de una aplicación implica reproducir su funcionalidad sobre una nueva plataforma tecnológica, en la que normalmente variarán el lenguaje de programación, el sistema operativo y los mecanismos de acceso a los datos.

Mucha de la funcionalidad que constituye una aplicación informática procede precisamente de los mecanismos soportados por el sistema operativo, más allá de aquélla procedente del lenguaje de desarrollo.

Se enfrenta así el problema de cómo reproducir, en las aplicaciones convertidas, funcionalidad compleja aportada por el sistema operativo origen en un sistema operativo destino que a veces no incluye dichas capacidades.

Esto resulta especialmente destacado en el caso de sistemas operativos que, como el OS/400, aportan una gran riqueza funcional y de gran especificidad.

Mecanismos tales como las colas, los procesos *submitidos* o la gestión de usuarios resultan un eslabón perdido que el técnico necesita cuando se pretenden convertir las aplicaciones desarrolladas en OS/400 hacia otro sistema operativo.

Caravel OS/400 FRAMEWORK ha sido diseñado para satisfacer esta necesidad. Para ello, ofrece implementada en clases 100% *pure* Java toda la funcionalidad del OS/400 para su uso en cualquier sistema operativo en el que ésta pueda ser requerida.

Construido como un conjunto de clases Java de utilización inmediata, su uso resulta totalmente similar al del OS/400 estándar.

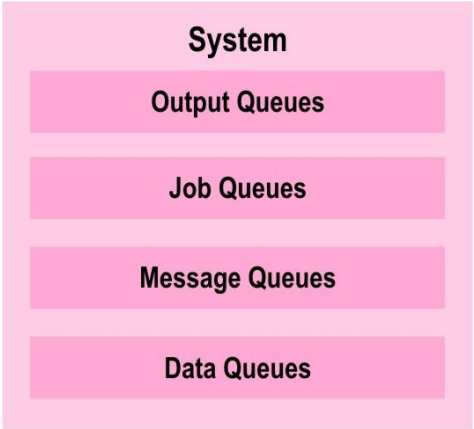
2. Funcionalidad soportada

Caravel OS/400 FRAMEWORK proporciona un entorno con las mismas características y utilidades principales que ofrece un sistema OS/400, y que permiten al usuario tener un mayor control sobre los recursos generados por una aplicación.

Para ello incluye un entorno que permite interactuar con la información del sistema, utilizando los mismos comandos que se utilizarían en un sistema OS/400 para su gestión.

Caravel OS/400 FRAMEWORK no es un entorno cerrado, sino que permite la definición de nuevos comandos que amplíen los ya existentes, posibilitando de esta forma la creación de un sistema totalmente personalizado a las necesidades del usuario.

Caravel OS/400 FRAMEWORK proporciona un conjunto de características equivalentes a las que ofrece un sistema OS/400, como son:



- Sistema de gestión de listados o datos de salida.
- Sistema de gestión de trabajos.
- Sistema de gestión de mensajes.
- Sistema de gestión de áreas de datos de intercambio.

De este modo se proporciona un soporte para control de los diferentes recursos que genere la aplicación, como los informes, mensajes o datos mediante un conjunto de colas.

Caravel OS/400 FRAMEWORK ofrece una API con un conjunto de funciones que permiten:

- Controlar las colas de almacenamiento en los diferentes sistemas de gestión, al poder crear nuevos elementos con las características deseadas, modificar dichas características o eliminarlas del sistema.
- Controlar la información almacenada en las colas mediante funciones que nos permitirán conocer los elementos que hay y obtenerlos para poder operar sobre ellos.

Utilizando las funciones de la API podremos hacer que desde los programas de nuestra aplicación se pueda enviar información a las colas de determinados sistemas, como pueda ser el sistema de Spool, así como desarrollar una interfaz de consulta y modificación de los componentes del sistema de manera totalmente integrada en la aplicación.

3. Users Manager

Al igual que un sistema OS/400, *Caravel OS/400 FRAMEWORK* necesita tener una relación de los usuarios que pueden acceder a él, permitiendo personalizar en cada caso cierta información.

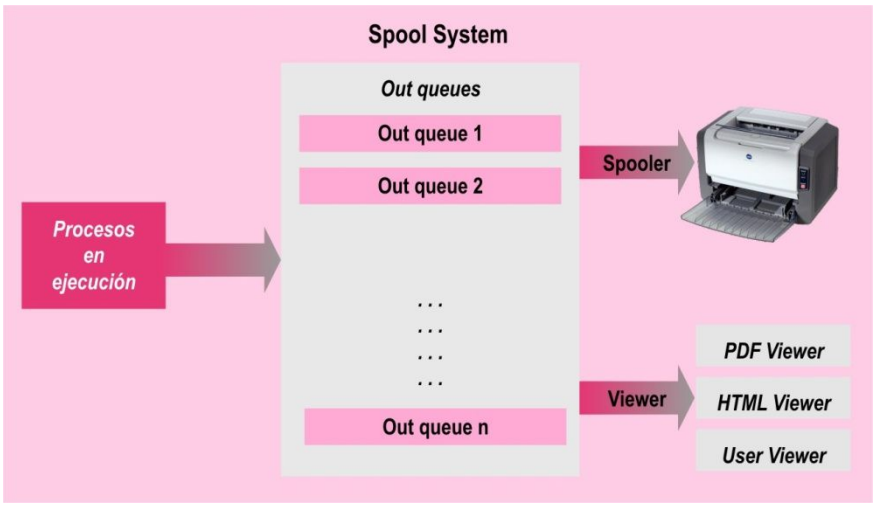
Cada usuario tendrá *"parametrizado"* qué recursos utilizará, como por ejemplo:

- Cola a la que mandará sus trabajos *"submitidos"*.
- Colas de mensaje por defecto que usará.
- *Spool* de impresión hacia el que enviará sus listados.

4. *Spool* System

Caravel OS/400 FRAMEWORK proporciona un *Spool* que permite colocar los listados en diferentes colas de impresión para posteriormente realizar algún tipo de operación con ellos (retenerlos, imprimirlos, visualizarlos, etc.).

En la gestión del *Spool* entran en juego varios elementos relacionados entre sí.



Los procesos en ejecución enviarán los informes generados a las colas de salida *(out queues)* del *Spool system.* Las colas de salida tienen dos estados:

- HOLD o retenida. Cualquier informe de la cola se considerará también retenido, aunque su estado sea diferente.
- READY o preparada. La cola se encuentra lista para ser procesada y permitirá al proceso *Spooler* que se lance y quiera consultarla que consulte su contenido para enviar los informes necesarios a la impresora.

Este estado podrá ser consultado o modificado mediante las funciones que proporcione la API.

Una cola de salida podrá estar relacionada con una impresora determinada, la cual puede recibir informes de diferentes colas. Esta relación puede ser configurada mediante funciones de la API. Cuando se envía un informe a la impresora, será el proceso *Spooler* el que se encargue de realizar la gestión necesaria.

Con la versión Java 1.3.1 un *Spooler* sólo puede imprimir sobre una impresora, con la versión 1.4 un mismo *Spooler* puede enviar informes a varias impresoras.

Los informes que estén en las colas, independientemente del estado que ésta pueda tener, pueden encontrarse en tres estados diferentes:

- HOLD. El listado queda en espera y no se imprime.

- READY. El listado se encuentra listo para ser impreso. El proceso *Spooler* que recorre las colas para ver qué elementos pueden ser procesados lo encontrará y lo enviará a la impresora asociada a la cola. Cuando termine, el listado desaparecerá de la cola.
- SAVE. El listado ya se ha impreso, pero se queda grabado.

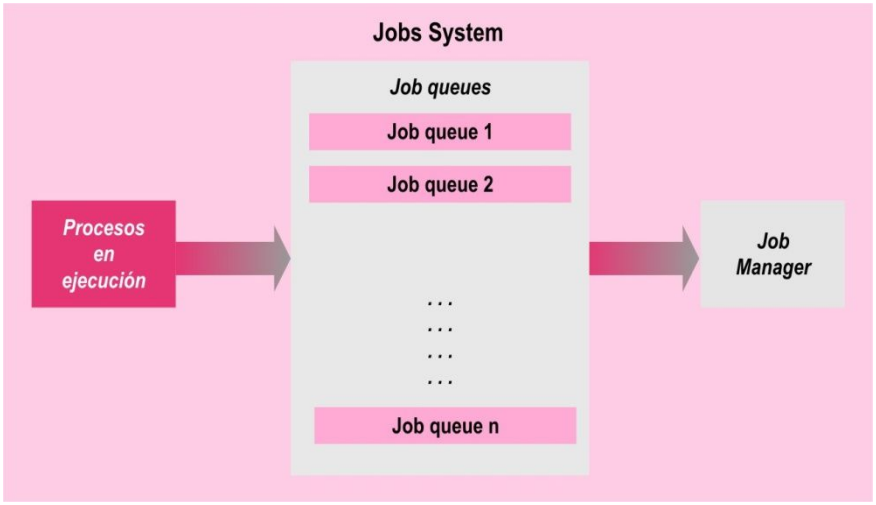
Es posible acceder a las colas y obtener la relación de elementos que la componen. También podemos interactuar con uno de ellos para:

- Modificar su estado.
- Enviarlo a la impresora asociada a la cola.
- Visualizarlo mediante el *viewer* correspondiente al formato requerido, ya sean los propios del *framework* o los creados a medida.
- Eliminarlo de la cola.

5. Jobs system

La ejecución de un programa o comando puede realizarse en diferido, es decir, es posible lanzar su ejecución ordenando que ésta no sea inmediata, sino que ocurra en un determinado momento o bajo la acción de un operador que la active.

Caravel OS/400 FRAMEWORK puede gestionar los trabajos ejecutados de esta forma de manera que se pueda decidir sobre cuándo se ejecutarán, modificar la prioridad sobre otros procesos de la cola, etc.



Los procesos en ejecución pueden invocar a otros de forma que su ejecución deba pasar por una cola de procesos en la que se almacene a la espera de la orden de ser ejecutados o que se cumpla una condición establecida al enviarlo a la cola, como pueda ser que se ejecute a determinada hora, o que un tercer proceso se encargue de cambiar su estado para que pueda ser ejecutado.

Se podrán enviar los procesos a cualquiera de las colas de trabajos que proporciona el sistema y que podremos crear y modificar mediante las funciones que proporciona la API.

Las colas de trabajos, como las de salida, tienen dos estados:

- HOLD o retenida. La cola se encuentra retenida, y por lo tanto también lo estarán los trabajos que contenga, aunque su estado sea diferente.
- READY o preparada. La cola permitirá al proceso *job manager* que consulte su contenido para poner en ejecución aquellos trabajos que sean necesarios.

Un *job* que se encuentra encolado tendrá un estado asociado:

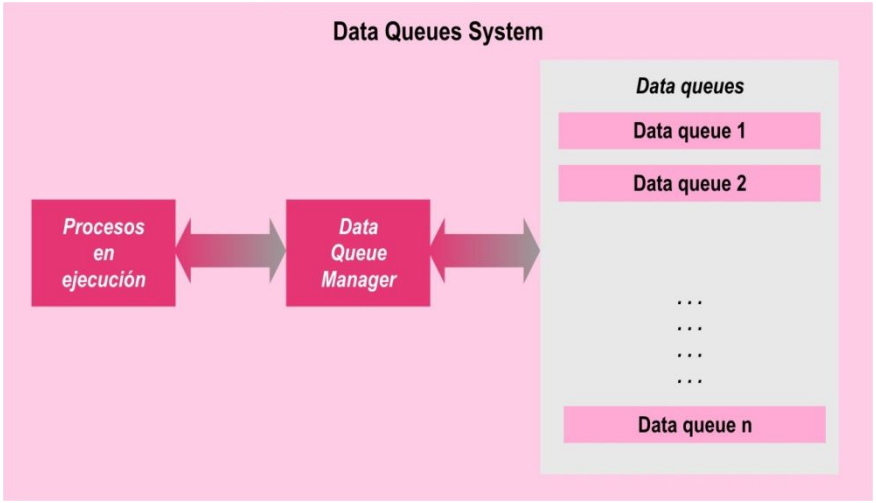
  - HOLD. El trabajo se encuentra retenido y no se ejecutará.
  - DECEASED. La ejecución del programa sufrió algún problema.
  - END. Ejecución terminada.

- SCHEDULED&HELD. La ejecución se encuentra retenida, ya que está programada para un determinado momento.
- INITIALIZED.
- QUEUED.
- ACTIVE.
- READY. Listo para ser ejecutado, el *job manager* lo extraerá de la cola y lo pondrá en marcha.

Será un proceso *job manager* quien se encargue de revisar el contenido de las colas de trabajos y verificar cuáles de los que se encuentran encolados deben ponerse en ejecución.

6. Data queue system

Las colas de datos permiten el paso de información entre programas, al proporcionarles un área estática de intercambio.

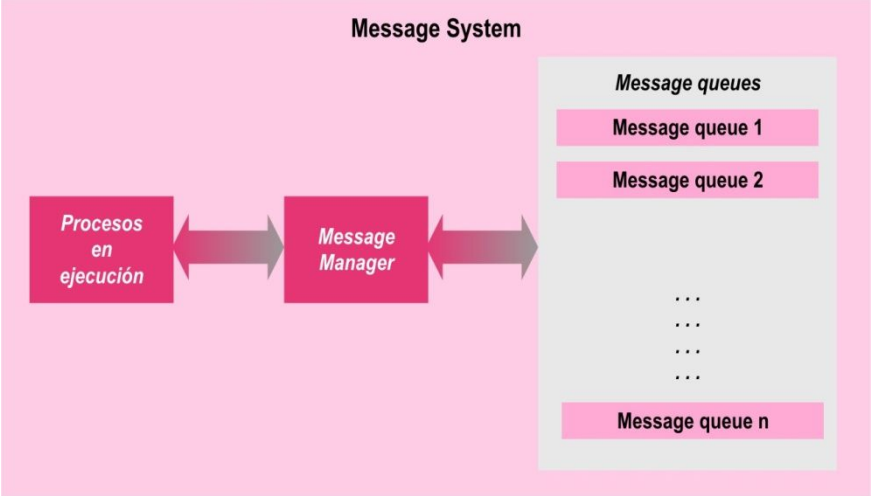


En el sistema de colas de datos se podrán definir colas con diferentes características, como el tipo de dato que almacena o su capacidad, de forma que los procesos serán capaces de compartir información de varios tipos enviándola a la cola que sea necesaria según la naturaleza de los datos que quiera enviar.

Mediante el *data queue manager* los procesos serán capaces de acceder a las colas de datos para recoger y enviar información.

7. Message System

En un sistema OS/400 los mensajes proporcionan una forma de comunicación especializada de las colas de datos entre los programas de una aplicación. Consecuentemente, *Caravel OS/400 FRAMEWORK* implementa esta característica.



Los procesos en ejecución de la aplicación, mediante la funcionalidad proporcionada por la API, serán capaces de enviar mensajes a las colas definidas en el sistema. Este sistema proporciona un mecanismo de comunicación entre programas, e incluso con los usuarios, ya que podrán tener acceso a las colas de mensajes.

Por ejemplo, en una ejecución nocturna de una serie de procesos, éstos podrán enviar mensajes a una cola propiedad del administrador del sistema y donde se le informe del estado de terminación de los procesos. El usuario administrador podrá consultar los mensajes almacenados en dicha cola e ir eliminando aquellos que no necesiten ser utilizados por otros procesos.

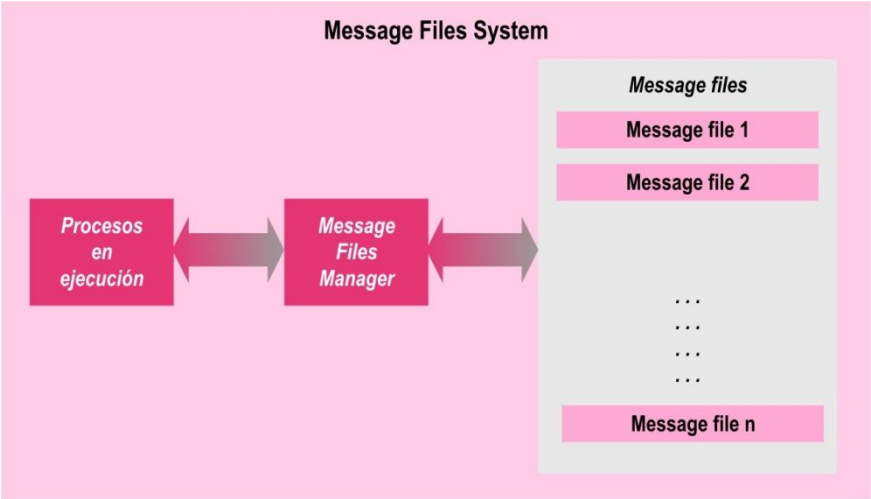
También se puede controlar la ejecución de determinados procesos condicionándolos a la existencia o no de un mensaje concreto en una cola del sistema.

El *message manager* proporcionará la funcionalidad necesaria para que los procesos puedan extraer los mensajes de las colas, así como añadir nuevos mensajes.

8. Message Files System

Los ficheros de mensajes permiten centralizar literales y mensajes frecuentemente utilizados, ofreciendo así una mayor facilidad de mantenimiento al limitarse la modificación de cualquier mensaje a modificar su entrada en el fichero al que pertenece.

Los ficheros de mensajes no pertenecen a la gestión de mensajes, pero en una cola de mensajes, en lugar de almacenar un literal cualquiera podremos almacenar una descripción de mensaje almacenada en un fichero de mensajes.



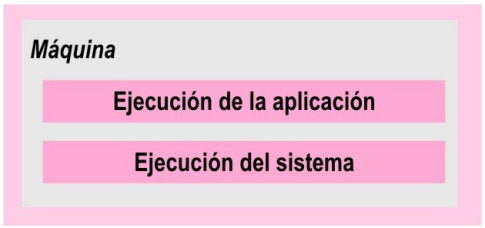
En lugar de utilizar literales estáticos definidos en un programa se puede acceder a un fichero de mensajes determinado para obtener ese literal, el cual podrá enviarse a una cola de mensajes, a un informe de impresora o mostrarlo por pantalla.

La API del *framework* proporciona también la funcionalidad necesaria para crear los ficheros de mensajes que necesitemos y crear las descripciones de mensajes que éstos almacenen.

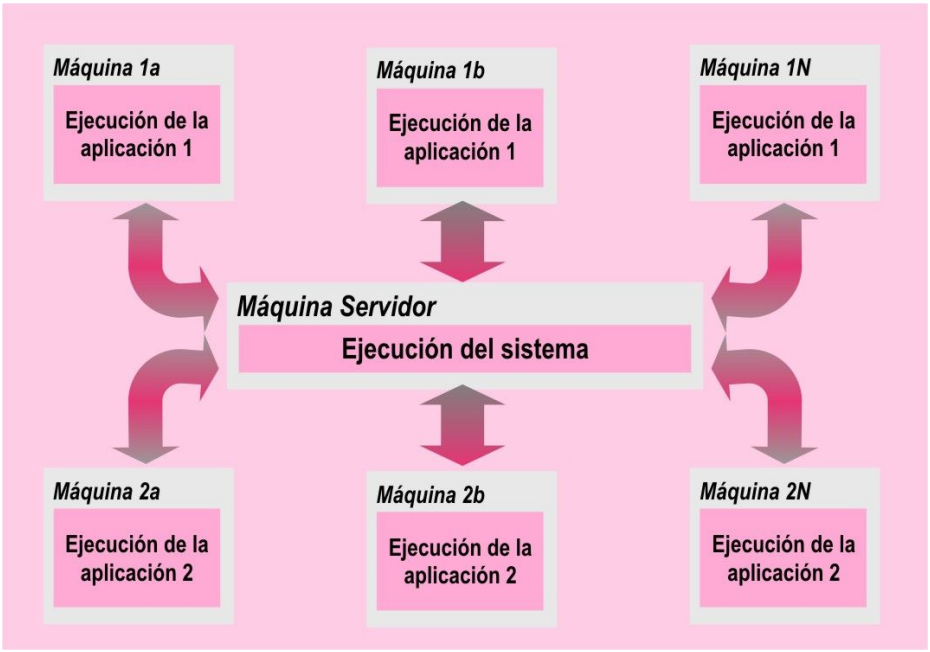
9. Arquitecturas de ejecución

Caravel OS/400 FRAMEWORK se apoya en la existencia de una base de datos en la que se almacenarán los componentes del sistema. Mediante las funciones de la API se accede a su contenido, de forma independiente a donde realmente esté situada.

- Sistema instalado en local. Las funciones de acceso al sistema se ejecutan en la misma máquina en la que se ejecuta la aplicación, y sólo ese puesto accede a la información del sistema. Aunque es posible esta configuración, no permite a otros puestos de trabajo acceder a la información de sistema.



- Sistema instalado en remoto. La aplicación se estará ejecutando en varias máquinas, las cuales accederán al sistema cuando lo necesiten, pero la funcionalidad de acceso al sistema se ejecutará únicamente en el servidor. En esta configuración se encontrará centralizada en un servidor, de forma independiente a cómo accedan a los datos de la aplicación. La funcionalidad de acceso al sistema se ejecuta en el servidor, tanto si los datos se encuentran en éste como en una maquina diferente.



El primer tipo de arquitectura es válido para la realización de pruebas en modo local y para desarrollo. La segunda configuración es la que realmente aprovecha *Caravel OS/400 FRAMEWORK*, que es capaz de proporcionar recursos a diferentes aplicaciones.