[Módulo 1. Introducción 1](#_Toc138178395)

[Introducción 1](#_Toc138178396)

[Grupos de Rutinas 1](#_Toc138178397)

[Cuadro Resumen 1](#_Toc138178398)

[Módulo 2. Rutinas de Arranque del Sistema 2](#_Toc138178399)

[Arranque del Sistema 2](#_Toc138178400)

[Función *FechaHoraUTC* en el Arranque 2](#_Toc138178401)

[Función *SemanaUTC* en el Arranque 2](#_Toc138178402)

[Módulo 3. Rutinas Horarias programadas 3](#_Toc138178403)

[Introducción 3](#_Toc138178404)

[Función *RutinasHorarias* 3](#_Toc138178405)

[Función *LinksEnProd* 3](#_Toc138178406)

[Función *MailDeFeedback* 4](#_Toc138178407)

[Función *ProdsEnRCLV* 4](#_Toc138178408)

[Módulo 4. Rutinas Diarias Programadas 5](#_Toc138178409)

[Introducción 5](#_Toc138178410)

[Función *FechaHoraUTC* 5](#_Toc138178411)

[Función *RutinasDiarias* 5](#_Toc138178412)

[Función *ImagenDerecha* 6](#_Toc138178413)

[Función *PaisesConMasProductos* 6](#_Toc138178414)

[Función *SemanaUTC* en las Rutinas Diarias 6](#_Toc138178415)

[Módulo 5. Rutinas Semanales Programadas 7](#_Toc138178416)

[Introducción 7](#_Toc138178417)

[Función *SemanaUTC* 7](#_Toc138178418)

[Función *RutinasSemanales* 7](#_Toc138178419)

[Función *AprobadoConAvatarLink* 7](#_Toc138178420)

[Función *BorraImagenesSinRegistro* 8](#_Toc138178421)

[Función LinksVencidos 8](#_Toc138178422)

[Función RclvsSinEpocaPSTyConAno 8](#_Toc138178423)

# Introducción

## Introducción

### Visión general

* Las rutinas sirven para:
  1. Corregir errores de la base de datos.
  2. Actualizar la información
  3. Generar outputs automáticos.
* Todas las rutinas y sus procesos principales, se encuentran en la carpeta de Funciones/Rutinas

### Corregir errores de la Base de Datos

* Estos se pueden ocasionar de diversas maneras, algunas de las cuales pueden ser:
  1. Situaciones que el sistema no llega a contemplar.
  2. Novedades introducidas manualmente en el sistema.

### Actualizar Información

* Hay información que se debe actualizar con alguna periodicidad, que puede ser horaria, diaria o semanal.

### Generar Outputs Automáticos

* Hay algunos outputs que se deben generar en forma automática, con cierta periodicidad.

## Grupos de Rutinas

* Se pueden ejecutar en alguna de tres periodicidades diferentes.
  1. Cada vez que arranca el sistema
  2. Horaria
  3. Diaria
  4. Semanal

## Cuadro Resumen

|  |  |  |
| --- | --- | --- |
| Rutina | Objetivo | Periodic. |
| *FechaHoraUTC* | Actualizar la información | Arranque |
| *RutinasDiarias* | Ejecutar las rutinas diarias | Arranque |
| *SemanaUTC* | Actualizar la semana en JSON | Arranque |
| *RutinasSemanales* | Ejecutar las rutinas semanales | Arranque |
| ***RutinasHorarias*** | **Ejecutar las rutinas horarias** | **Horaria** |
| *LinksEnProd* | Actualizar la información | Horaria |
| *MailDeFeedback* | Generar outputs automáticos | Horaria |
| *ProdsEnRCLV* | Actualizar la información | Horaria |
| ***RutinasDiarias*** | **Ejecutar las rutinas diarias** | **Diaria** |
| *FechaHoraUTC* | Actualizar la información | Diaria |
| *ImagenDerecha* | Generar outputs automáticos | Diaria |
| *PaisesConMasProductos* | Actualizar información | Diaria |
| ***RutinasSemanales*** | **Ejecutar las rutinas semanales** | **Semanal** |
| *AprobadoConAvatarLink* | Corregir errores de la BD | Semanal |
| *BorraImagenesSinRegistro* | Corregir errores de la BD | Semanal |
| *LinksVencidos* | Actualizar información | Semanal |
| *RCLV\_idEnCapitulos* | Corregir errores de la BD | Semanal |
| *RCLVsSinEpocaPSTyConAno* | Corregir errores de la BD | Semanal |
| *SemanaUTC* | Actualizar la información | Semanal |

# Rutinas de Arranque del Sistema

## Arranque del Sistema

### Introducción

* Cuando arranca el sistema, necesitamos tener seguridad de que las rutinas necesarias fueron ejecutadas a su debido tiempo.
* Para verificarlo, se hace un repaso por c/u de ellas, mediante la función ***startupMasConfiguracion***.

### Función *startupMasConfiguracion*

* Realiza 2 tareas fundamentales:
  1. Configura la periodicidad de ejecución de las rutinas horarias y diarias.
  2. Ejecuta la función ***FechaHoraUTC***.
* Las rutinas horarias se ejecutarán solamente cuando sea el horario configurado, sin importar cuándo se ejecutaron por última vez antes de que arrancara el sistema.
* Para las rutinas diarias se usa un criterio distinto, se ejecutan apenas arranca el sistema.

## Función *FechaHoraUTC* en el Arranque

* Lo primero que hace es averiguar en qué fecha se ejecutó por última vez. Si se ejecutó ese mismo día, interrumpe la función.
* En caso contrario,
  1. Actualiza las *FechaUTC* y *HoraUTC* por las vigentes.
  2. Ejecuta la función ***RutinasDiarias***, que a su vez ejecuta todas las demás rutinas diarias, una a continuación de la otra.
  3. Ejecuta la función ***SemanaUTC***.

## Función *SemanaUTC* en el Arranque

* Lo primero que hace es averiguar en qué semana se ejecutó por última vez. Si se ejecutó esa misma semana, interrumpe la función.
* En caso contrario,
  1. Actualiza la *SemanaUTC* por la vigente. Usa la fecha “UTC”. Cada semana cambia los días domingo.
  2. Ejecuta la función ***RutinasSemanales***, que a su vez ejecuta todas las demás rutinas semanales, una a continuación de la otra.

# Rutinas Horarias programadas

## Introducción

* La única función horaria programada es ***RutinasHorarias***.
* Ella se ocupa de ejecutar las demás.

## Función *RutinasHorarias*

* Objetivo: ejecutar las rutinas horarias:
  1. LinksEnProd
  2. MailDeFeedback
  3. ProdsEnRCLV
* Tareas:
  1. Obtiene la información del archivo JSON
  2. Actualiza todas las rutinas horarias
* Funciones que usa:
  1. procesos.lecturaRutinasJSON
  2. this.LinksEnProd
  3. this.MailDeFeedback
  4. this.ProdsEnRCLV

## Función *LinksEnProd*

* Objetivo: actualizar el status de los productos, respecto de sus *links*, *links gratuitos*, y en idioma *castellano*.
* Tareas por cada registro de *películas* y *capítulos*
  1. Obtiene los ID de los registros.
  2. Por cada ID, ejecuta la función ***linksEnProd***.
* Función ***linksEnProd***:
  1. Averigua si para el registro de producto existe c/u de éstos: *link, linkGratuito, castellano, subtitulos*
  2. En cada caso, las respuestas pueden ser: SI (con el valor 2), talvez (con el valor 1), NO (con el valor *null*).
  3. Actualiza el registro original del producto, en c/u de esos campos.
* Tareas por cada registro de *colecciones*
  1. Obtiene los ID de los registros.
  2. Por cada ID, ejecuta la función ***linksEnColec***.
* Función ***linksEnColec***:
  1. Averigua si para el registro de colección, sus capítulos son mayoría (≥ 50%) de SI, talvez, NO, para c/u de éstos: *link, linkGratuito, castellano, subtitulos*
  2. Actualiza el registro original de la colección, en c/u de esos campos.
* Funciones que usa:
  1. procsCRUD.revisiones.linksEnProd
  2. procsCRUD.revisiones.linksEnColec
  3. procesos.finRutinasHorarias

## Función *MailDeFeedback*

* Objetivo: informarles a los usuarios el resultado de sus sugerencias de Data Entry, según fueron revisadas por nuestro equipo.
* Al ser esta función más compleja que las demás, las tareas y funciones usadas se explican en un módulo dedicado.
* En condiciones normales de operación del sistema, cada usuario recibirá el mail una sola vez por día, en un horario cercano a las 0hs del país en el que informó su residencia.

## Función *ProdsEnRCLV*

* Objetivo: actualizar en cada registro de RCLV, si tiene productos vinculados.
* Tareas por cada entidad:
  1. Obtiene los ID de los registros de la entidad
  2. Por cada ID, ejecuta la función ***ProdsEnRCLV***
* Función ***ProdsEnRCLV***
  1. Si el rclv\_id es ≤ 10, termina la función
  2. Establece la condición perenne, que es que el campo *rclv\_id* del producto, debe coincidir con el ID del registro RCLV.
  3. Respuesta SI: averigua si existe algún producto en status *aprobado*, con ese rclv\_id.
  4. Respuesta talVez: else, averigua si existe algún producto en status *provisorio*, con ese rclv\_id
  5. Respuesta talvez: else, averigua si existe alguna edición con ese rclv\_id
  6. Respuesta NO: No encontró ningún caso
  7. Actualiza el campo en el RCLV con la respuesta

# Rutinas Diarias Programadas

## Introducción

* La única función diaria programada es ***FechaHoraUTC***.
* Ella se ocupa de ejecutar las demás si corresponde.

## Función *FechaHoraUTC*

* Objetivos:
  1. Actualizar las FechaUTC y HoraUTC por las vigentes.
  2. Que se ejecuten las rutinas diarias.
  3. Si corresponde, que se ejecuten las rutinas semanales.
* Tareas:
  1. Obtiene la información del archivo JSON
  2. Obtiene la fecha y hora UTC actual
  3. Actualiza las *FechaUTC* y *HoraUTC* por las vigentes.
  4. Actualiza los campos de Rutinas Diarias.
  5. Verifica si se deben correr las Rutinas Semanales.
* Funciones que usa:
  1. procesos.lecturaRutinasJSON
  2. procesos.fechaHoraUTC
  3. procesos.guardaArchivoDeRutinas
  4. procesos.finRutinasDiariasSemanales
  5. this.RutinasDiarias
  6. this.SemanaUTC

## Función *RutinasDiarias*

* Objetivo: ejecutar las rutinas diarias:
  1. ImagenDerecha
  2. PaisesConMasProductos
* Tareas:
  1. Obtiene la información del archivo JSON
  2. Actualiza todas las rutinas diarias
* Funciones que usa:
  1. procesos.lecturaRutinasJSON
  2. this.ImagenDerecha
  3. this.PaisesConMasProductos

## Función *ImagenDerecha*

### Visión general

* Objetivo: actualizar la *Imagen Derecha* estándar que se ve en las vistas.
* Tareas:
  1. Limpia el historial de *ImagenesDerecha* en *global*.
  2. Actualiza los títulos de la imagen derecha para cada fecha y descarga las imágenes nuevas.
  3. Guarda los títulos de las imágenes nuevas.
  4. Borra los archivos de imagen que no se corresponden con los títulos.
* Funciones que usa:
  1. procesos.lecturaRutinasJSON
  2. procesos.diaMesAno
  3. procesos.***obtieneImgDerecha***
  4. comp.gestionArchivos.copiaImagen
  5. procesos.guardaArchivoDeRutinas
  6. procesos.borraLosArchivosDeImgDerechaObsoletos
  7. procesos.finRutinasDiariasSemanales

### Función *obtieneImgDerecha*

* Objetivo:
* Tareas:
  1. Obtiene el *diaDelAno\_id* de la fecha que se necesita
  2. Obtiene los RCLV, mediante la función ***obtieneLosRCLV***
  3. Filtra por los que tienen la máxima prioridad\_id
  4. Elige al azar de entre los que tienen la máxima prioridad
  5. Si se encontró un solo resultado, lo asigna
  6. Obtiene los datos de la imgDerecha, mediante la función *datosImgDerecha*
* Si se encontró un RCLV, los datos que devuelve son:
  1. entidad, id
  2. título: el apodo y si no tiene el nombre
  3. carpeta y nombreArchivo
* Si no se encontró un RCLV, los datos que devuelve son:
  1. título: "ELC - Películas"
  2. carpeta y nombreArchivo

## Función *PaisesConMasProductos*

* Objetivo: actualizar la cantidad de productos realizados en cada país, para identificar el TOP 5 que se muestra con prioridad cuando se elige el país de origen.
* Tareas:
  1. Obtiene la frecuencia por país
  2. Actualiza la frecuencia por país
* Función que usa: procesos.finRutinasDiariasSemanales

## Función *SemanaUTC* en las Rutinas Diarias

* Lo primero que hace es averiguar en qué semana se ejecutó por última vez. Si se ejecutó esa misma semana, interrumpe la función.
* En caso contrario,
  1. Actualiza la *SemanaUTC* por la vigente. Usa la fecha “UTC”. Cada semana cambia los días domingo.
  2. Ejecuta la función ***RutinasSemanales***, que a su vez ejecuta todas las demás rutinas semanales, una a continuación de la otra.

# Rutinas Semanales Programadas

## Introducción

* No hay ninguna función semanal programada, porque si el día que corresponde no se ejecuta, no se ejecuta en toda la semana.
* Por esa razón, se optó ejecutar en forma diaria la función ***SemanaUTC***, que revisa si cambió la semana y en ese caso se ocupa de ejecutar las demás rutinas semanales si corresponde.

## Función *SemanaUTC*

* Objetivos:
  1. Actualizar la *SemanaUTC* por la vigente.
  2. Que se ejecuten las Rutinas Semanales.
* Tareas:
  1. Obtiene la información del archivo JSON.
  2. Obtiene la fecha y hora UTC actual.
  3. Obtiene la semanaUTC actual.
  4. Si la *semanaUTC* actual es igual a la del archivo JSON, termina la función.
  5. Actualiza la *SemanaUTC* por la vigente.
  6. Actualiza los campos de Rutinas Semanales.
* Funciones que usa:
  1. procesos.lecturaRutinasJSON
  2. procesos.fechaHoraUTC
  3. procesos.semanaUTC
  4. procesos.guardaArchivoDeRutinas
  5. procesos.finRutinasDiariasSemanales
  6. this.SemanaUTC

## Función *RutinasSemanales*

* Objetivo: ejecutar las rutinas semanales:
  1. AprobadoConAvatarLink
  2. BorraImagenesSinRegistro
  3. LinksVencidos
  4. RclvsSinEpocaPSTyConAno
* Función que usa: procesos.lecturaRutinasJSON

## Función *AprobadoConAvatarLink*

* Objetivo: corregir el error si algún registro original aprobado (Productos y RCLVs) tiene un url en vez de un nombre de archivo en el campo avatar. Lo descarga y guarda en la carpeta 'Prods-Final' correspondiente a cada familia.
* Tareas:
  1. Revisa el campo avatar en cada registro de cada entidad, excepto *capítulos*.
  2. Descarga el archivo
  3. Actualiza el valor en el campo avatar
* Función que usa: procesos.finRutinasDiariasSemanales

## Función *BorraImagenesSinRegistro*

* Objetivo: corregir el error, cuando encuentra un archivo de imagen que no está registrado en el campo avatar de ninguna entidad, en las carpetas:
  1. Final
  2. Revisar
  3. Provisorio
* Tareas:
  1. Elimina las imágenes sin registro original ni edición, en Productos.
  2. Elimina las imágenes sin registro original ni edición, en RCLVs.
  3. Elimina las imágenes obsoletas (más de 3 días), en Provisorios.
* Funciones que usa:
  1. procesos.eliminaImagenesDeFamiliasSinRegistro
  2. procesos.borraImagenesProvisorio

## Función LinksVencidos

* Objetivo: propone revisar nuevamente, los links aprobados hace más de un tiempo prudencial.
* Tareas:
  1. Obtiene la fecha de corte
  2. Establece que las condiciones de los links vencidos son que la fecha del *statusSugeridoEn* sea menor que la *fecha de corte*, y que el *statusRegistro\_id* sea *aprobado*.
  3. Prepara la información a guardar: statusSugeridoPor\_id: 2, statusRegistro\_id: creadoAprob\_id
  4. Actualiza el status de los links vencidos
* Función que usa: procesos.finRutinasDiariasSemanales

## Función RclvsSinEpocaPSTyConAno

* Objetivo: corregir el error de que hayan quedado registros RCLV (personajes y hechos) con una época distinta a *pst*, y con el campo *ano* con algún valor.
* Tareas:
  1. Fija que las condiciones son que el *statusRegistro\_id* sea *aprobado*\_id, que la *epoca\_id* sea distinta de *pst*, y que el *ano* sea distinto de *null*.
  2. Busca los registros de personajes y hechos que cumplan con esa condición.
  3. Los actualiza, llevando a *null* el valor del campo *epoca\_id*.
* Función que usa: procesos.finRutinasDiariasSemanales

# Función *MailDeFeedback*

## Tareas

* Obtiene de la base de datos, la información de todo el historial pendiente de comunicar, y si no hay pendientes interrumpe. Para todo esto, usa la función ***obtieneElHistorial***.
* Une en una sola variable todos los registros. Si no hay registros a comunicar, termina el proceso.
* Obtiene los usuarios relacionados con esos registros
* Por cada usuario realiza una rutina, con varias tareas.
* Mientras realiza la rutina para cada usuario, avisa que está procesando el envío de los mails.
* Ejecuta la función ***finRutinasHorarias***.

## Rutinas por Usuario

* Obtiene la fecha local del usuario, según la zona horaria del país donde nos comunicó su residencia. Usa la función ***hoyUsuario***.
* Si la fecha local del usuario es igual que la fecha del último comunicado enviado a él (campo *fechaRevisores*), se saltea el usuario.
* Arma el cuerpo del mail, con las funciones
  1. ***mensajeStatus***
  2. ***mensajeEdición***.
* Envía el mail al usuario y actualiza la BD, mediante las funciones:
  1. ***enviarMail***
  2. ***eliminaRegsStatusComunica***
  3. ***eliminaRegsEdicComunica***
* Guarda en el registro del usuario, la fecha en que se le envió el mail, en el campo *fechaRevisores*.

## Funciones que usa

### Función *obtieneElHistorial*

#### Visión General

* Objetivo: obtener todos los registros del feedback que se le debe dar a cada usuario.
* Ruta: procesos.mailDeFeedback.obtieneElHistorial

#### Tareas

* Obtiene los registros de *histStatus*.
  1. Condiciones: tiene que ser un resultado de revisión (*aprobado* distinto a *null*), y no haber sido comunicado antes (*comunicadoEn* igual a *null*).
  2. Los registros con el campo aprobado igual a *null*, son cambios de status realizados por un usuario, sin revisión.
* Obtiene los registros de *histEdics*. La única condición es no haber sido comunicado antes (*comunicadoEn* igual a *null*).
* A todos los registros se les agrega el campo *tabla*, con el valor que corresponda entre:
  1. histStatus
  2. histEdics

### Función *hoyUsuario*

* Objetivo: devuelve la fecha local del usuario.
* Ruta: procesos.finRutinasHorarias

### Función *mensajeStatus*

* Objetivo: elaborar el mensaje a enviarle al usuario, por los cambios de status.
* Ruta: procesos.mailDeFeedback.mensajeStatus
* Tareas:
  1. De cada registro, obtiene los campos clave o los elabora
  2. Ordena la información según los campos de mayor criterio, siendo el primero la familia y luego la entidad
  3. Crea el mensaje en formato texto para cada registro de status, y se lo asigna a *mensajesAprob* o *mensajesRech*
  4. Crea el *mensajeGlobal*, siendo primero los aprobados y luego los rechazados
  5. Devuelve el *mensajeGlobal*

### Función *mensajeEdición*

* Objetivo: elaborar el mensaje a enviarle al usuario, por los cambios de edición.
* Ruta: procesos.mailDeFeedback.mensajeEdición
* Tareas:
  1. De cada registro, obtiene los campos clave o los elabora
  2. Ordena la información según los campos de mayor criterio, siendo el primero la familia y luego la entidad
  3. Crea el mensaje en formato texto para cada entidad, y sus campos editados
  4. Crea el mensajeGlobal

### Función *enviarMail*

* Objetivo: enviar el mail al usuario
* Ruta: comp.enviarMail

### Función *eliminaRegsStatusComunica*

#### Visión General

* Objetivo: eliminar los registros prescindibles del *histStatus*, y en los demás completar el campo de *comunicadoEn*.
* Ruta: procesos.mailDeFeedback.eliminaRegsStatusComunica

#### Tareas:

* Establece la combinación de opciones que se deben eliminar:
  1. statusOriginal\_id: *creado\_id*, statusFinal\_id: *creadoAprob\_id*
  2. statusOriginal\_id: *creadoAprob\_id*, statusFinal\_id: *aprobado\_id*
  3. statusOriginal\_id: *creado\_id*, statusFinal\_id: *aprobado\_id*
* Elimina los registros en los que se cumple alguna de las condiciones.
* A los demás registros, les completa el campo *comunicadoEn*.

### Función *eliminaRegsEdicComunica*

* Objetivo: eliminar los registros prescindibles del *histEdic*, y en los demás completar el campo de *comunicadoEn*.
* Ruta: procesos.mailDeFeedback.eliminaRegsEdicComunica
* Tareas:
  1. Estable que el criterio para borrar registros es que el campo duración sea *null* o tenga valor cero.
  2. Elimina los registros en los que se cumple la condición.
  3. A los demás registros, les completa el campo comunicadoEn.