

CAPITULO SEIS

6 AGENTE ADMINISTRADOR

La funcionalidad del agente administrador es la de apoyar al administrador de BIDACI, indicándole el material agregado por los académicos. El agente administrador genera un reporte semanal de las aportaciones de cada académico para llevar un control del acervo.

El agente administrador notifica al administrador de BIDACI acerca de aquellos usuarios (estudiantes) que no han entrado al ambiente, enviando un correo electrónico de advertencia para poder permanecer en BIDACI. El agente administrador es capaz de eliminar las claves de los estudiantes en caso de no responder al llamado.

El agente administrador notifica al administrador de BIDACI los cambios en el material educativo, al revisar los sitios del material que sigan siendo accesibles. En caso de que el material tenga problemas para visitarla, el agente administrador se encargará de enviar correos electrónicos al autor del material notificando que tiene problemas de acceso (ver figura 6.1).

Además el agente administrador notifica por correo electrónico al académico de la falta de actualización de su material, en base un periodo de tiempo (180 días), analizando el tamaño y la fecha del material registrado en el acervo.

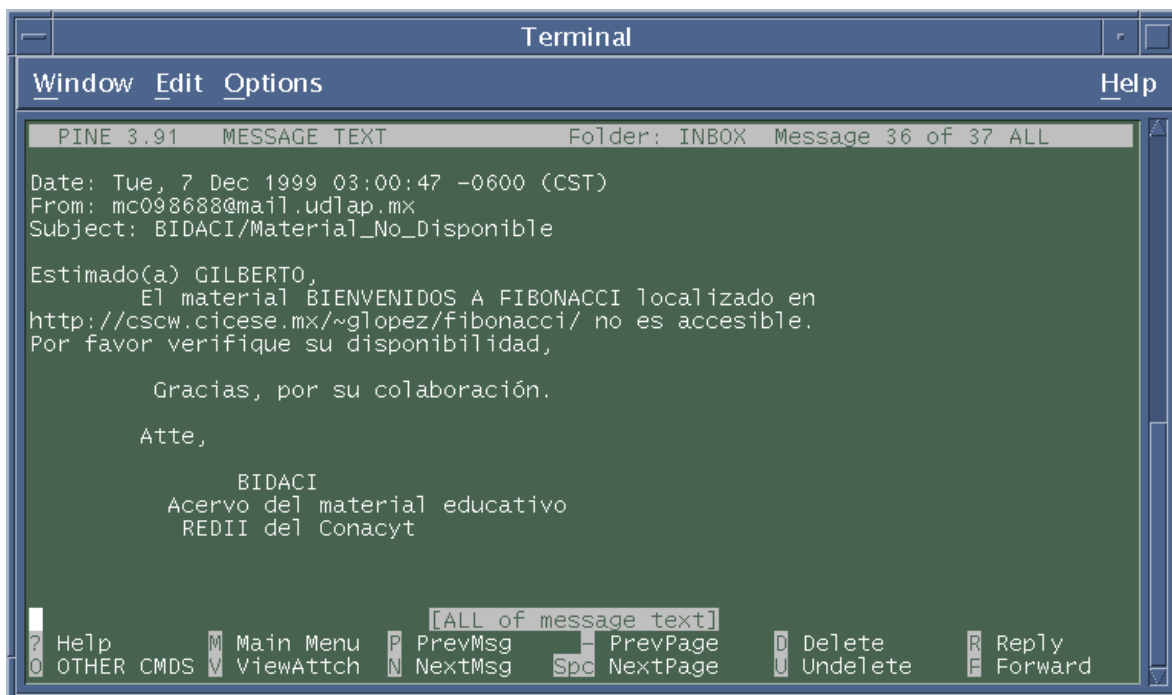


Figura 6.1 Notificación de las ligas al académico.

Por otro lado, si el agente administrador detecta la ausencia de un estudiante por un periodo de tiempo de 180 días, le envía a éste tres mensajes de advertencia (uno cada tres días), recordándole su nombre de usuario y clave para entrar al ambiente (ver figura 6.2). De no responder, el estudiante será dado de baja, ya que se considera que su visita al ambiente fue sólo por una investigación o tarea que se le encargó durante éste período, además de que puede desanimar a los estudiantes que lo estén invitando a formar un grupo de discusión, al no responder al llamado.

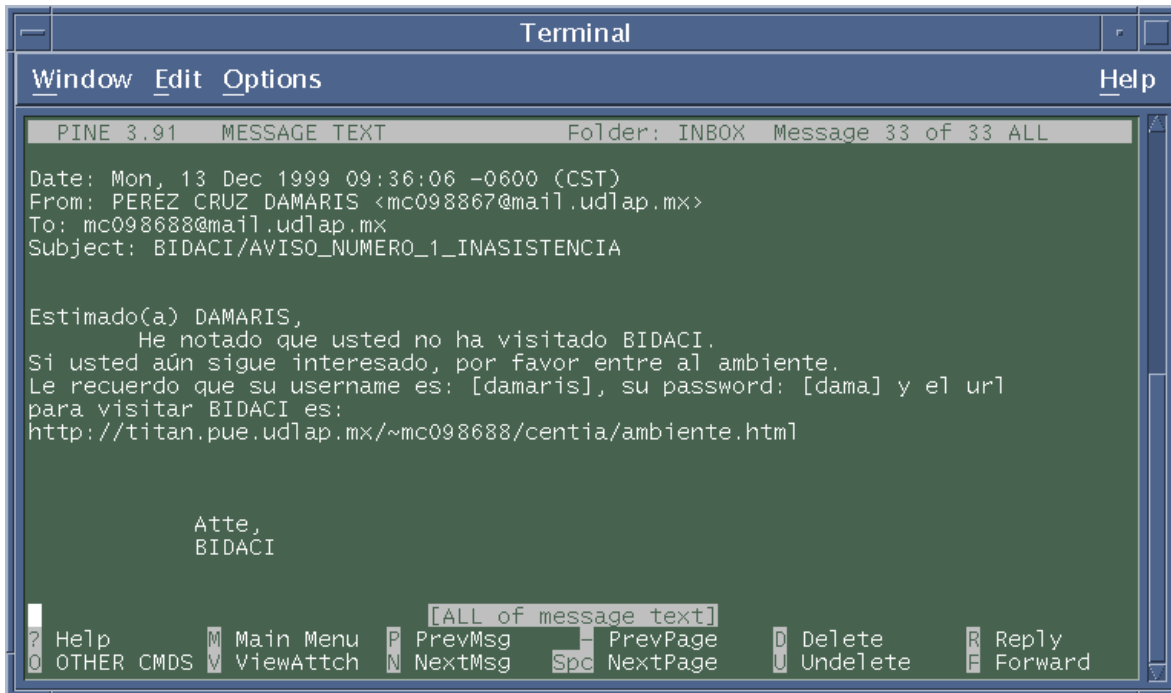


Figura 6.2 Notificación de las inasistencias al estudiante.

Las tareas que lleva a cabo el agente administrador se ejecutan de manera concurrente con el ambiente. La concurrencia se logra a través de hilos de ejecución (Threads), los cuales permiten ejecutarse en paralelo con otros hilos. Cada thread (hilo, flujo de control del programa) representa un proceso individual ejecutándose en un sistema. A veces se les llama procesos ligeros o contextos de ejecución. Cada thread controla un único aspecto dentro de un programa, como puede ser revisar las ligas del material que son accesibles, controlar las inasistencias de los estudiantes, verificar la actualización del material agregado por los académicos y generar un reporte de las aportaciones del material educativo.

6.1 Modelo del Agente Administrador

En la tabla 3 se presenta el modelo del agente administrador:

| Clases | Atributos | Métodos |
|---------------------|---|--|
| AgenteAdministrador | username_usuario, email_usuario, titulo_acervo, url_acervo, tamaño_acervo, fecha_alta_acervo, fecha_entrada_usuario, MaterialEducativo, EmailCliente. | obtenerDatosDelUrl, revisarAsistenciaEstudiantes, verificarEntradaEstudiante, generarReporte. |
| MaterialEducativo | url_académico, url_titulo, nombre_académico, email_academico, tamaño_documento. | revisarUrlNoDisponible, revisarActualizaciónUrl compararTamañoUrl EmailCliente |
| EmailCliente l | Para, tittulo, mensaje | sendMensaje. |

Tabla 6.3 Estructura del agente administrador

La función del agente administrador es cumplir ciertas tareas específicas en la cual fue programado como el de enviar correos electrónicos para notificar la falta de asistencia del

estudiante, notificar al autor las ligas que no son accesibles y la continua actualización del material. Además de generar un reporte semanal de las aportaciones de los académicos.

En la figura 6.3 se muestra la arquitectura del agente administrador en donde se logra la participación con el usuario vía correo electrónico, así como sus acciones para el cual fue diseñado.

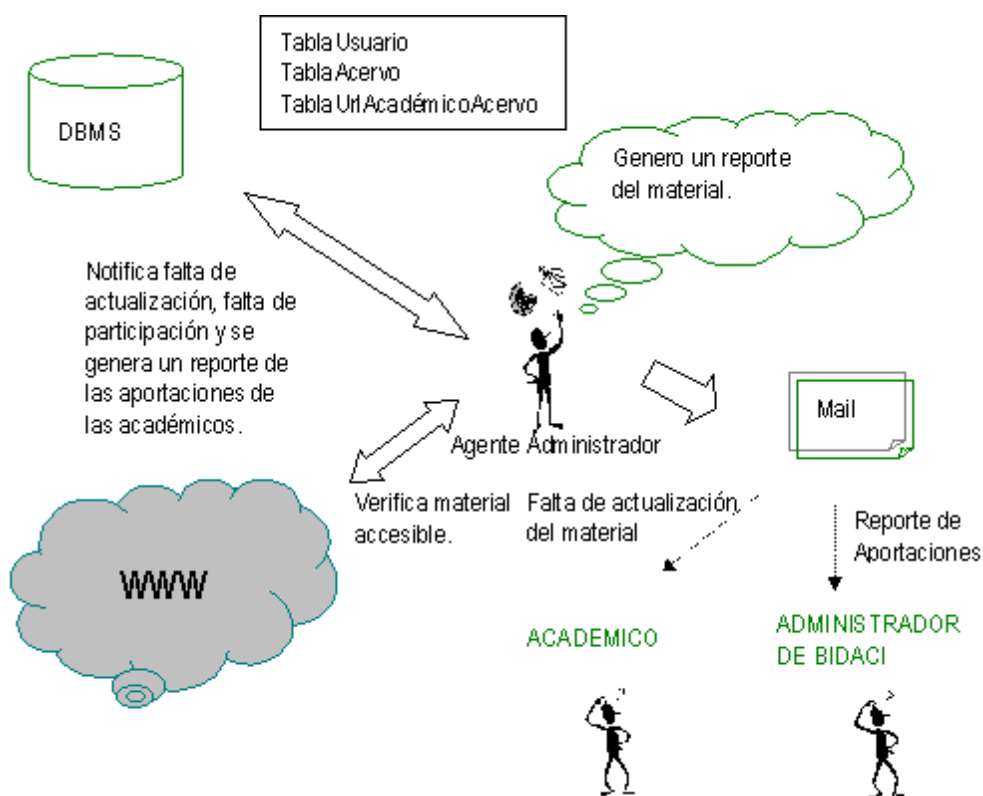


Figura 6.3 Arquitectura del agente administrador

6.2 DESCRIPCIÓN DEL MODELO DEL AGENTE ADMINISTRADOR.

Clase implementada:

AgenteAdministrador

Esta clase es una interfaz Runnable y contiene los siguientes atributos:

username_usuario,

email_usuario,

título_acervo,

url_acervo,

fecha_alta_acervo,

tamaño_acervo,

fecha_entrada_usuario.

La implementación de la interfaz Runnable nos permite tratar nuestra clase como un objeto Runnable. Al igual que cuando derivamos de la clase Thread, el código que controla el hilo se coloca en el método run.

Los métodos implementados son:

obtenerDatosDelUrl.

Este método obtiene de la base de datos todas las ligas, títulos, académico y la fecha de alta del documento.

verificarEntradaEstudiante

Este método obtiene la fecha de entrada de todos los estudiantes y verifica si tiene algún aviso de advertencias.

revisarAsistenciaEstudiantes.

Este método verifica el tiempo de inasistencias con la fecha entrada del estudiante y la fecha actual y calcula los días que no ha entrado al ambiente. También utiliza la

clase EmailCliente para enviar los correos electrónicos de advertencias a los estudiantes.

generarReporte.

Este método genera un reporte semanal de las aportaciones de los académicos.

EmailCliente.

Esta es una clase que realiza las conexiones al servidor de socket desde el Applet.

RevisarMaterialEducativo.

Esta clase contiene los siguientes atributos:

url_académico,
url_titulo,
nombre_académico,
email_academico,
tamaño_documento

Los métodos implementados son:

revisarUrlNoDisponible.

Este método realiza las conexiones a los site de las páginas para verificar su existencia, en caso de tener problemas, envía un correo electrónico al académico, utlizando el método sendEmail de la clase EnviarUnEmail.

revisarActualizaciónUrl

Este método calcula los días basándose en la fecha de alta del acervo, para poder enviar correos electrónicos los académicos notificando su actualización.

compararTamañoUrl

Este método compara el tamaño en byte de cada documento con los del acervo para determinar si ya fue actualizado.

EnviarUnEmial

Esta clase contiene los siguientes atributos:

para,

título,

mensaje,

Su principal método es:

sendMensaje.

Permite realizar las conexiones por sockets al servidor de correos electrónicos y enviar mensajes.