



## APÉNDICE C. MÁS SOBRE AGLETS WORKBENCH

El API Aglet es un conjunto de clases e interfaces escritas en Java que permiten la implementación de agentes móviles. Esta herramienta fue creada por el Laboratorio de Investigación de IBM Tokyo en respuesta a las necesidades de proveer una plataforma uniforme para los agentes móviles que se encuentran en ambientes heterogéneos como Internet.. La Herramienta de Desarrollo de Software Aglets (ASDK por sus siglas en inglés) es una implementación del API Aglet e incluye paquetes, documentación, ejemplos, aglets y el servidor de aglets Tahiti (aplicación que permite al usuario enviar, manipular y recibir aglets a otras computadoras que estén corriendo Tahiti), todo esto bajo el Protocolo de Transferencia de Agentes (ATP)

Entre las principales clases e interfaces que integran el J-AAPI (Java Aglet Application Programming Interface) se encuentran las siguientes:

Clases.

**Aglet.** Es el corazón del API Aglet ya que en ella se definen los métodos para controlar el ciclo de vida del aglet: métodos para clonar, crear, despachar, retraer, activar y desactivar al propio aglet. Esta clase también permite acceder a los atributos específicos de cada agente, esto por medio del objeto llamado AgletInfo.

**Message.** La comunicación entre los aglets se lleva a cabo por medio del intercambio de objetos de esta clase, pudiendo enviar mensajes futuros, y recibir o no respuesta. Esta clase puede manejar mensajes con prioridad, mensajes pararelos, así como interrupción y destrucción de mensajes.

**AgletID.** A cada aglet se le asigna una identidad única a nivel global que se conserva a través de su ciclo de vida. El objeto identificador es inmutable y se puede recuperar directamente del proxy del aglet.

**FutureReply.** Esta interfaz permite regresar mensajes en forma asíncrona. Con ella quien lo recibe puede determinar si la respuesta está disponible y esperar por los resultados un tiempo determinado

para después continuar su ejecución en caso de que la contestación no haya sido regresada después del tiempo especificado.

## Interfaces.

**AgletContext.** El contexto del aglet es equivalente a su lugar de ejecución, ya que es aqu' donde el aglet se crea, duerme y se destruye; cuando el aglet viaja en la red, éste se mueve de contexto en contexto. En general el contexto es un ambiente de ejecución uniforme para los aglets. El agente usa la interfaz AgletContext para solicitar información sobre su ambiente para enviarle mensajes, incluyendo a otros aglets que se encuentran activos en ese contexto.

**AgletProxy.** Actúa como manejador del aglet y provee de un acceso seguro a los métodos públicos de los agentes. Por lo anterior si un aglet quiere comunicarse con otro, primero debe de obtener el proxy y después interactuar con él a través de esta interfaz..

[sinopsis](#) [índice](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [A](#) [B](#) [C](#) [D](#) [E](#) [bibliografía](#)

Chevalier Dueñas, G. A. 2000. Agentes móviles para la recuperación personalizada de información. Tesis Licenciatura. Ingeniería en Sistemas Computacionales. Departamento de Ingeniería en Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas Puebla. Mayo.

Derechos Reservados © 2000, Universidad de las Américas Puebla.

