



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA

ALGORITMOS Y PROGRAMACIÓN II (75.41)

Cátedra Ing. Patricia Calvo

Cuatrimestre Primero, 2010

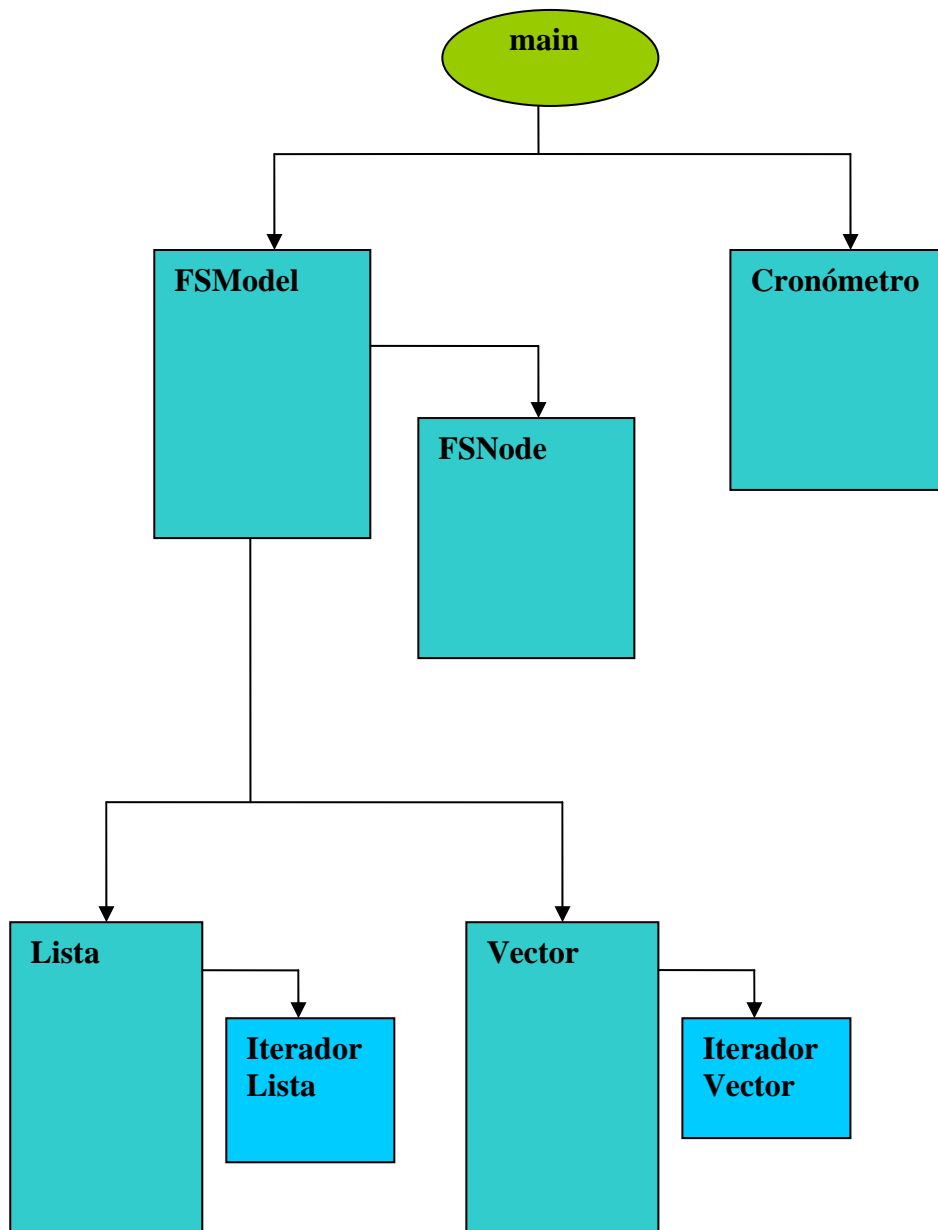
Trabajo Práctico Primero:
Analizador de Archivos

Grupo 4

Integrantes:

Apellido y Nombre	Padrón
Churca, Federico	91533
Fernández, Mariano	89789
González Lisella, Fiona	91454
Muñoz Catalán, Martín	87876
Samá, Nicolás	89456

Diagrama de relación entre las clases



DESCRIPCION GENERAL:

TDA FSMODEL: La clase fsmodel es la que interactúa directamente con el main. Es la encargada mediante sus métodos de cargar, volcar los datos y buscar las expresiones ingresadas por el usuario.

Utiliza contenedores genéricos (lista y vector) y sus operaciones básicas.

El método load se encarga de recolectar direcciones de archivos y directorios, los ordena y almacena como fsnodes en el contenedor.

El método search filtra las expresiones de entrada, se puede buscar por nombre de archivo o directorio, y se puede acortar búsquedas por tamaño.

Para la búsqueda invoca a las funciones booleanas MatchesExpression que comparan un patrón con una entrada.

TDA FSNODE: fsNode es un objeto utilizado por la clase fsModel.

Contiene la información que describe un archivo o un directorio.

Sus métodos se encargan de cargar los atributos y de devolver los mismos.

Atributos:

- **isdir:** Booleano que indica con true si es directorio o false si es archivo.
- **nombre:** Refiere al nombre local del objeto.
- **Path:** Lugar donde se encuentra el archivo/directorio.
- **Fullname:** Concatenación de nombre y path.
- **Size:** En el caso de que el objeto se archivo toma el tamaño en bytes de este, si es directorio se suma el contenido de los archivos y subdirectorios que posee.
- **mtime:** Indica la fecha de la última modificación.

TDA LISTA - TDA VECTOR: Lista y vector son las estructuras de datos genéricas que se utilizan para llevar a cabo el programa. En nuestro caso las usamos con fsModel. La lista es simplemente enlazada con un puntero al principio y otro al final, y el vector es dinámico que comienza con un tamaño de una posición y cada vez que requiere de mas espacio dobla el tamaño, crea un nuevo vector con esta especificación y copia el contenido del anterior.

Tanto la lista como el vector utilizan iteradores.

Las estructuras cumplen el mismo rol y son necesarias para medir el tiempo que utilizan al realizar las operaciones de búsqueda, mediante el cronometro se calcula dicho tiempo.

TDA CRONOMETRO: Este TDA fue implementado por la cátedra, mide los tiempos que transcurren en ciertos intervalos.

Tiene un contador del tipo clock_t y un booleano para saber si esta pausado o no. Sus métodos son iniciar, pausar, continuar y parar. Serán llamados desde el main.

BIBLIOGRAFIA CONSULTADA:

De Wikipedia:

- C POSIX library http://en.wikipedia.org/wiki/C_POSIX_library
- Dired.h <http://en.wikipedia.org/wiki/Dired.h>
- Stat [http://en.wikipedia.org/wiki/Stat_\(Unix\)](http://en.wikipedia.org/wiki/Stat_(Unix))

The C++ Resources Network

- Informacion General <http://www.cplusplus.com/>