Programación en Entornos Interactivos.
Enunciado Prácticas
febrero-julio 2013.

D.L.S.I.

febrero 2013

Índice

1. Enunciado.

Realizar dos versiones de una misma aplicación[1] empleando en cada versión de la práctica:

- una de las bibliotecas siguientes: Qt y Gtkmm.
- la *arquitectura MVC* para estructurar el código fuente de la misma. El código del *Modelo* será **obligatoriamente** el mismo en las dos prácticas y, para que quede claro, se encontrará en un único directorio aparte, como se puede ver en la figura 1.
- IMPORTANTE:

  Esta estructura de directorios es la que ***obligatoriamente*** tendrás que emplear para entregar el código de las prácticas.

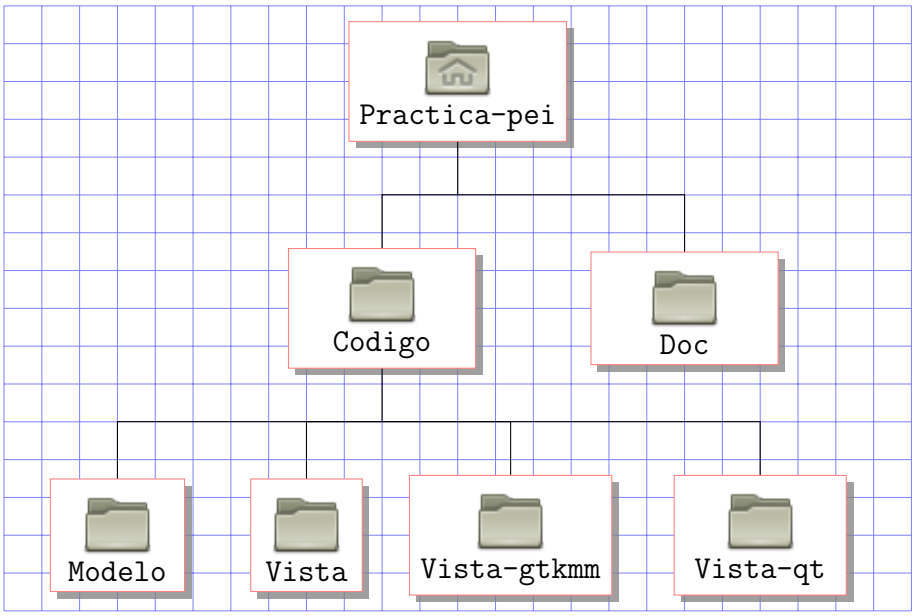

Figura 1: Estructura jerárquica de directorios con el código fuente y documentación de la práctica.

Esta jerarquía de directorios constará de un fichero `Makefile` por cada directorio. El `Makefile` del directorio `Practica-pei` se encargará de descender por cada subdirectorio y ejecutar `make` allí. El resultado final será un

[1] Debe compilar y ejecutarse con el software instalado en los laboratorios asignados para las prácticas de la asignatura.

fichero ejecutable en el subdirectorio `Vista-gtkmm` y otro en `Vista-qt`, evidentemente, cada uno de ellos enlazado con su biblioteca correspondiente.

1.1. Cometido de la aplicación.

Esta aplicación, **presentando un interfaz de usuario coherente y fácil de utilizar**[2], implementará un gestor de datos sencillo.

Esta aplicación será capaz de leer y guardar archivos de texto estructurados en registros con el siguiente formato:

```
NOMBRE    Juan Perez Lopez
DIRECCION
POBLACION Alicante
CPOSTAL   E3465
TELEFONO
EMAIL     jpl@correo.com

NOMBRE    nnn
DIRECCION ddd
POBLACION ppp
CPOSTAL   cpc
TELEFONO  tfn
EMAIL     eml

...
```

Como se puede ver el archivo consta de:

- cero o más registros, separados uno del siguiente por una línea en blanco.
- cada registro consta de 6 filas. Cada fila es un campo del registro. Siempre están en el mismo orden, como se puede ver en el ejemplo.
- cada fila consta de dos columnas, la primera es el nombre del campo (siempre en mayúsculas) y la segunda el valor que tiene en este registro. El valor puede estar vacío como se puede ver en el ejemplo.

Una vez leído uno de estos archivos mediante la opción correspondiente -p.e. eligiendo esta opción de un menú- la aplicación, presentando un interfaz de usuario coherente y fácil de usar, permitirá:

[2]Se puntuará (positiva o negativamente) el uso de los widgets más adecuados en cada caso.

1. Visualizar uno a uno los registros leídos del fichero. Para ello mostrará todos los datos del registro "actual" y además permitirá...

2. ... movernos por los registros del fichero hacia adelante o atrás de uno en uno.

3. Buscar uno o varios registros por cualquiera de sus campos. En este caso los registros visualizados por los que nos podremos "mover" serán los que cumplan esta condición. Evidentemente siempre podremos volver a mostrar todos los registros del fichero actual como al principio cuando se lee.

4. Modificar cualquier dato del registro que se esté visualizando en un momento determinado.

5. Borrar el registro actual que se esté visualizando.

6. Crear un nuevo registro y darle valores a sus campos.

7. Guardar en el archivo original el estado actual de los registros.

   Nota: El programa debe de tener una opción "**Acerca de...**" que nos presente el nombre de los *autores* de la práctica, y las *mejoras incluidas* en la misma.

2. Optativo.

Se dejan como optativas las siguientes características:

- Disponer de una opción `Guardar como...` que permita al usuario elegir un fichero distinto al original para guardar los datos posiblemente *modificados* en memoria.

- En el caso de la búsqueda se podrá:

  - Indicar si se busca una coincidencia exacta de cada cadena en un campo de un registro o se busca una subcadena.
  - Indicar si en el caso de la búsqueda con valores en varios campos de un registro estos se "conectan" con un "`OR`" o con un "`AND`".

- Poder trabajar con varios ficheros abiertos simultáneamente de manera que eligiendo de algún modo el fichero que nos interesa de entre los abiertos, visualicemos sólo sus registros.

El cambio entre ficheros se podrá hacer tantas veces como el usuario quiera y entre todos los ficheros que tenga abiertos la aplicación.

En todo momento debe quedar claro para el usuario con qué fichero está trabajando en un momento dado.

- Exportar a formato `CSV` pudiendo elegir el carácter separador entre campos, de este modo podremos importar estos archivos en un hoja de cálculo.
- Desarrollar la práctica con la biblioteca XForms compartiendo la parte del modelo con las otras dos versiones.

3. Mejoras

El programador podrá hacer las mejoras que considere oportunas, por ejemplo:

- Implementar la opción '`Nuevo fichero`', la cual permitirá crear registros en un fichero que no existe hasta que se crea con esta opción.
- Comprobar si un fichero está mal formado...por ejemplo, que a un registro le falta algún campo, o que tiene un campo cuyo nombre no se corresponde con los soportados.
- Poder movernos entre registros no sólo de uno en uno sino de '`n`' en '`n`'.
- Cualquier otra que te parezca interesante y útil.

4. Documentación.

En tres ficheros de texto ASCII llamados:

1. '`documentacion_modelo.txt`'
2. '`documentacion_vista_qt.txt`'
3. '`documentacion_vista_gtkmm.txt`'

y localizados en el subdirectorio '`Doc`' de la figura 1 debes indicar:

- Las funciones, métodos y clases que has creado, explicando qué hace cada una de ellas y para qué sirve cada uno de sus parámetros.
- El criterio que has seguido para dividir el código fuente en los distintos ficheros que utilizas.

5. Plazo, formato, entrega y avisos.

1. La entrega de la convocatoria de 'febrero' se realizará durante los **días del 27 de mayo al 2 de junio de 2013**. La entrega de la convocatoria de 'julio' se realizará durante los **días del 1 al 7 de julio de 2013**. Si alguna de estas fechas tuviera que ser cambiada se indicaría mediante un aviso en el Campus Virtual.

   IMPORTANTE:

   > Este es el enunciado de la práctica de julio al que hay que añadirle como obligatorias las partes optativas excepto la relativa a la versión de 'xforms'.

2. Se deberá entregar un fichero '.tgz' con nombre '`pei.tgz`' que contendrá la estructura de directorios de la figura 1.

   Dentro de los directorios oportunos estarán:

   - Los ficheros con el código fuente apropiado.
   - Los ficheros con la documentación.
   - Un fichero '`Makefile`' que facilite la compilación de cada versión.
   - El fichero o ficheros creados con '`qt-designer`' y/o '`glade`'.

   Dentro del subdirectorio inicial 'Practica_pei' habrá un fichero llamado '**datos.txt**' que contendrá el nombre de los alumnos.

3. La entrega **sólo** se realizará vía Web[3] en la siguiente página.

4. **No se corregirán aquellas prácticas que**:

   - No se hayan realizado con las versiones de las bibliotecas instaladas en los laboratorios de prácticas.
   - No se hayan realizado con las especificaciones explicadas en clase (por ejemplo: se debe emplear `gtkmm` en lugar de `gtk+`, cargando dinámicamente el archivo con la descripción del interfaz creada con `glade` y **no** generando el código `C/C++` del interfaz).
   - Los archivos 'Makefile' deben funcionar en los laboratorios de prácticas y los programas compilarán sin errores.

[3] No se aceptará ninguna entrega hecha por correo electrónico u otro medio distinto al especificado.