

Capítulo

3

Conclusiones

1. Conclusiones Finales

Este proyecto me ha permitido ponerme en contacto con una tecnología emergente como es OSGi. Son muchos los fabricantes, tanto en el mundo de la automoción como de la electrónica, que han apostado por esta tecnología para hacer llegar a sus dispositivos los servicios que se ofrecen en la red, por lo que el futuro a medio plazo de esta tecnología parece asegurado.

He adquirido gran experiencia en el desarrollo de aplicaciones en plataforma Linux, y en el manejo del entorno de desarrollo Eclipse 3.0.

Sin embargo, al tratarse de una tecnología tan novedosa he encontrado muchas dificultades a la hora de encontrar apoyo para desarrollar aplicaciones para esta tecnología.

Con respecto a los objetivos marcados para nuestro proyecto, se ha conseguido implementar en un alto porcentaje el propósito inicial del mismo.

Se ha conseguido definir la interfaz java del servicio de interfaz de usuario de forma clara, y se han realizado dos implementaciones que aseguran el correcto funcionamiento de dicho servicio.

Hemos conseguido generar el bundle de la aplicación tal y como planteamos en los objetivos iniciales del proyecto. Hemos realizado pruebas para verificar el correcto funcionamiento del mismo en un pc con Linux en el que estaba instalado el framework de OSGi.

Como ejemplo de aplicación hemos generado un tutorial en forma de aplicación gráfica donde podemos ver los distintos “widgets” de la vista swing y de la vista texto.

CONCLUSIONES

Podemos ver aquí como una misma descripción XUL puede ser mostrada en ambas vistas manteniendo la estructura definida por el desarrollador de la interfaz gráfica:

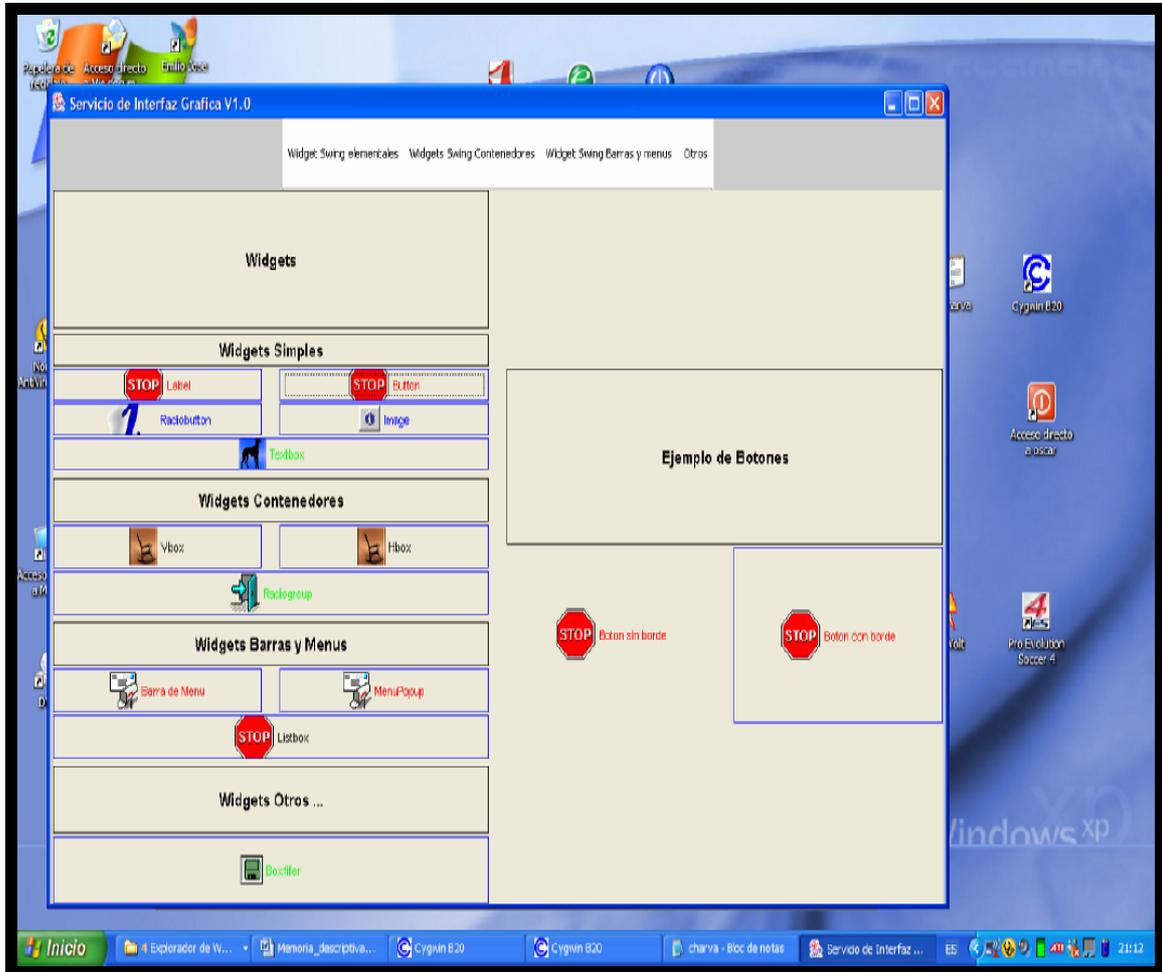


Figura 16: Vista Swing del SwingTextTutorial.

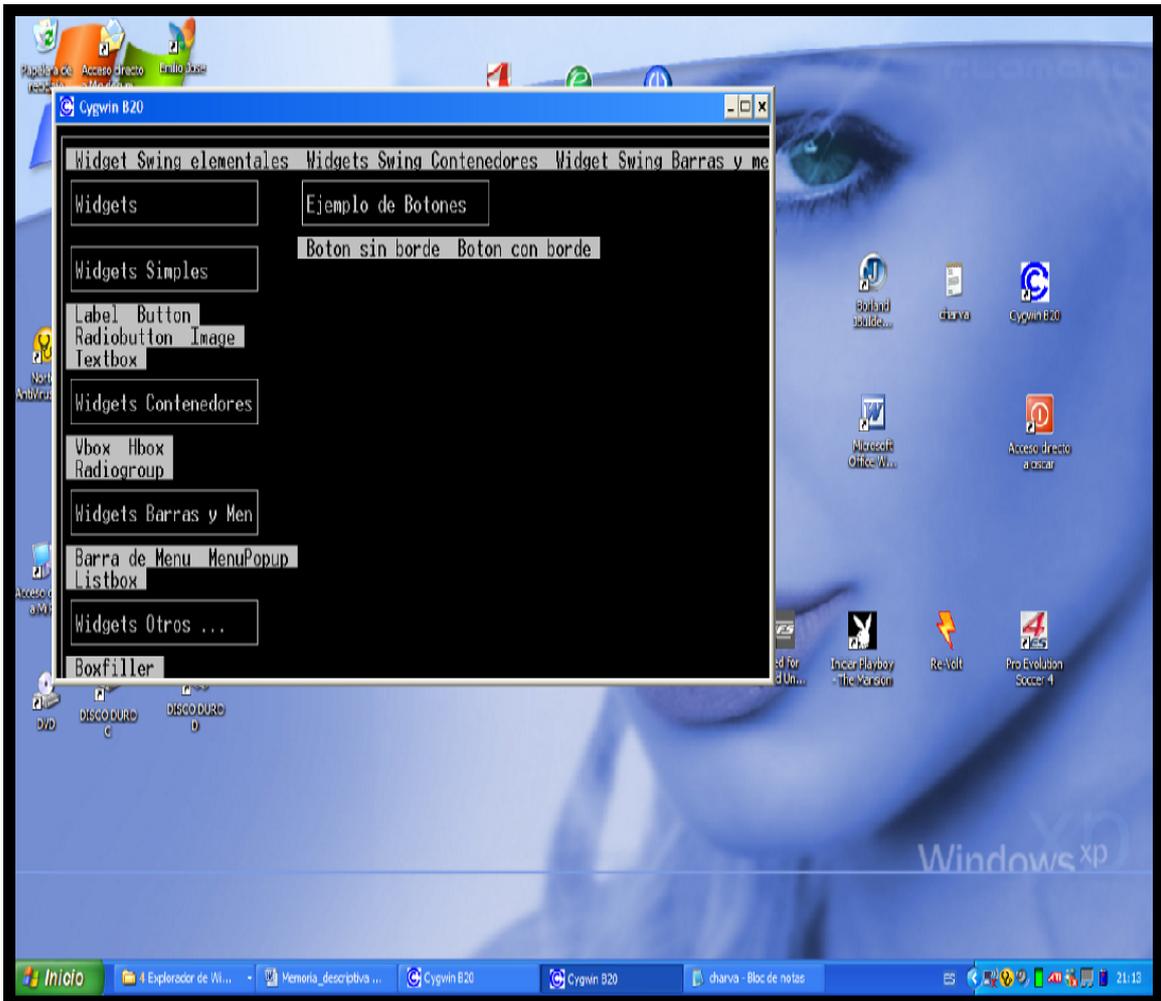


Figura 17. Vista Texto del SwingTextTutorial.

2. Ampliaciones y Posibles mejoras

Son numerosas las posibles ampliaciones que podemos hacer a nuestro proyecto para aumentar su funcionalidad y su utilidad como sistema de desarrollo de interfaces de usuario.

- **Ampliación del número de vistas:** Hemos implementado sólo dos vistas de entre todas las posibles para el servicio de interfaz de usuario. Una de las vistas más útiles que quedaría por implementar sería una vista html. Se podrían usar etiquetas html para cada uno de los widgets y usar servlets de java para implementar los manejadores de evento.
- **Aumentar el número de widgets por vista:** Sería importante para poder usar el servicio de interfaz de usuario, tener un conjunto mucho más amplio de widgets, con el fin de facilitar la tarea al desarrollador de interfaces de usuario. Son muchas las posibilidades que quedarían por implementar como barras de progreso, estructuras en forma de árbol, ...
- **Realizar la modificación del lenguaje LXUL** para compatibilizarlo completamente con el lenguaje XUL. Una posible mejora sería buscar la compatibilidad del lenguaje LXUL con el lenguaje XUL. Esto haría que cualquier diseño de interfaz realizado para el servicio de interfaz de usuario pueda ser visualizado en el navegador mozilla.
- **Aumentar el número de métodos modificadores por Widget en cada una de las vistas:** Se han implementado sólo unos pocos métodos modificadores por widget con el fin de comprobar el correcto funcionamiento de la interfaz. Faltaría aumentar el número de estos métodos por Widget.
- **Implementación del módulo de detección de las capacidades gráficas del dispositivo:** Para automatizar el proceso de selección de vista, sería importante implementar un módulo que detecte las capacidades gráficas del dispositivo donde vayamos a representar el interfaz de usuario y elija la mejor opción de entre todas las vistas posibles.
Hasta ahora este proceso es manual, y es el programador quien elige a través de parámetros el tipo de “vista” en la que quiere que se represente su interfaz su usuario.
- **Por último proponemos como posible ampliación, la incorporación de un módulo que realice la gestión del espacio de la pantalla.** Es decir, este nuevo módulo arbitraría entre las distintas aplicaciones desplegadas en la pantalla de un dispositivo, manejando el espacio existente para la correcta visualización de las distintas aplicaciones.