|  |  |  |
| --- | --- | --- |
| **Objetivos**   1. Recordar el proceso de carga de información al mundo desde archivos de datos 2. Generar reportes de información del mundo en archivos de datos. 3. Practicar el refresco de información en la interfaz de la aplicación luego de cargar información nueva | **Número de participantes**   * **Juan Ortega López** * **Elkin Salcedo Benavides** | **Materiales**   * Internet * Eclipse |
| **Duración de la actividad**  **01:30** |
| **INSTRUCCIONES**   1. Ingresa a <https://cupi2.virtual.uniandes.edu.co/images/APO2/N8/CasosEstudio/Discotienda/n8_discotienda.zip>, descarga el ejercicio DiscoTienda, crea el proyecto en eclipse, ejecútalo y verifica que todas las opciones trabajan correctamente. **No olvides activar los assert en Eclipse.** 2. Se requiere un servicio en la disco tienda que permita desde un archivo de texto (.txt) cargar un listado de discos a la disco tienda. Cada disco debe tener nombre, artista, genero (pop, salsa, etc) y la ruta de la imagen del disco.    1. Complete el requerimiento funcional.  |  |  |  |  | | --- | --- | --- | --- | | ID: | RF1 | Nombre: | ListadoDiscos | | Resumen: | | | Este servicio permitirá cargar un listado de discos a la disco tienda desde un archivo de texto (.txt). Cada disco deberá tener un nombre, artista, género y la ruta de la imagen correspondiente. | | Entradas: | | | Un archivo de texto (.txt) que contenga la lista de discos a cargar, con la siguiente información para cada disco:  Nombre: el nombre del disco  Artista: el nombre del artista o banda que interpreta el disco  Género: el género musical al que pertenece el disco (por ejemplo, pop, salsa, rock, etc.)  Ruta de imagen: la ruta donde se encuentra la imagen del disco | | Resultado: | | | Una vez cargado el listado de discos, estos estarán disponibles en la disco tienda para su compra. Cada disco aparecerá con su nombre, artista, género y la imagen correspondiente. |  * 1. Cree un archivo de texto con la información de los discos a cargar en la disco tienda y guárdelo en el directorio /data de la aplicación   2. En la clase Discotienda programe el método que permite cargar del archivo creado en el literal anterior los discos. Utilice el método agregarDisco de la clase Discotienda como parte de la solución. No olvide anunciar las excepciones que se puedan generar.   3. Utilice el metodo1() para probar el anterior método programado, informando al usuario que se han cargado exitosamente los discos. No olvide atrapar las excepciones que se generan.   4. En la clase InterfazDiscotienda del paquete uniandes.cupi2.discotienda.interfaz ubique el método asociado con el botón Opción 1, y luego de mostrar el mensaje al usuario, refresque el combo de discos del panelDiscos llamando al método **public void**refrescarDiscos( ArrayList discos ) enviándole como parámetro la contenedora de discos de la disco tienda.   5. Ejecuta la aplicación y verifica el funcionamiento de tu solución.  1. En la disco tienda se requiere un servicio que permita generar un reporte de los discos con sus correspondientes canciones, de aquellos discos que tengan menos de un número determinado de canciones dado por el usuario, el archivo debe almacenarse en el directorio /data de la aplicación y debe tener el nombre de reporteDeDiscos.txt, de acuerdo con el siguiente formato:   **REPORTE DE DISCOS CON MENOS DE ### CANCIONES**  **-----------------------------------------------------------------------**  **DISCO: @@@ ARTISTA: @@@**  **CANCIONES:**  **-----------------**  **Nombre Precio Unidades Vendidas**  **--------------------------------------------------------------------------------------------**  **@@@@@@ ######### ###**  **…**  **--------------------------------------------------------------------------------------------**  **DISCO: @@@ ARTISTA: @@@**  **CANCIONES:**  **-----------------**  **Nombre Precio Unidades Vendidas**  **--------------------------------------------------------------------------------------------**  **@@@@@@ ######### ###**  **…**  **--------------------------------------------------------------------------------------------**  Se debe reemplazar los símbolos @@@ y ### por la información correspondiente   * 1. Programa el método que resuelve el problema planteado en la clase correspondiente. No olvides anunciar las excepciones que se pueden presentar.   2. Utiliza el metodo2() para probar el funcionamiento del anterior método con un mínimo de 3 canciones.   No olvides atrapar las excepciones que se puedan presentar.   * 1. Ejecuta la aplicación y verifica el correcto funcionamiento de tu solución.  1. Incluye este documento debidamente diligenciado en la carpeta docs/specs de la aplicación, comprime tu ejercicio y envíalo por la plataforma virtual. | | |