

Alan Ezequiel Pucci – TP3

Empresa de soporte técnico

El programa fue diseñado para una empresa de soporte técnico que requiere de una plataforma donde cargar la computadora que trae el cliente por reparar y la misma registre todo el flujo desde que se cargó hasta que se devolvió.

Los distintos estados que va a tener una computadora son (en orden):

Recibida, Por reparar, Reparada, Por entregar, Devuelta

Existen dos tipos de usuario: Recepcionista y técnico.

RECEPCIONISTA:

Username: recepcionista

Password: 12345

TÉCNICO:

Username: tecnico

Password:12345

La pantalla principal del recepcionista cuenta con:

- un panel de usuario para modificar sus datos,
- una lista con todas las computadoras cargadas,
- la posibilidad de cargar la computadora de un cliente.

El recepcionista es quien va a cargar la computadora para reparar, seleccionando los componentes que tiene la computadora y dejando una breve descripción.

Una vez cargada, va a poder visualizar todas las computadoras en sus distintos estados, por defecto va a estar seleccionada la opción de “Recibida” pero se puede modificar desde el Check Box ubicado en la esquina superior derecha.

Dentro de la lista de “Recibidas” el recepcionista va a tener la posibilidad de Modificar, Eliminar o Enviar al técnico la computadora para que el mismo la repare.

Una vez la computadora fue enviada al técnico, el recepcionista debe esperar a que este último se la devuelva reparada, para que luego pueda entregarla al cliente.

La pantalla principal del técnico cuenta con:

- un panel de usuario para modificar sus datos,
- una lista con todas las computadoras cargadas,
- NO tiene la posibilidad de cargar computadoras.

Dentro de la pantalla de “Listar computadoras” del técnico, va a poder visualizar únicamente las computadoras “Por reparar” y “Reparadas”, por defecto va a estar

seleccionada la opción de “Por reparar” pero se puede modificar desde el Check Box ubicado en la esquina superior derecha.

El técnico podrá reparar la computadora modificando alguno de sus componentes y cambiando la descripción. Una vez reparada, la computadora se va a situar en la lista de “Reparadas”, por lo que el técnico tendrá que seleccionar esa lista desde el Check Box y ENVIARLA al recepcionista (Si la computadora fue reparada pero no enviada al recepcionista, este último no podrá devolverla al cliente).

El flujo termina cuando el recepcionista recibe la computadora reparada y se la devuelve al cliente.

El recepcionista se encuentra guardado dentro de:
.../FrmPrincipal/bin/Debug/recepcionist.xml
De todas formas los valores por defecto son:
username: recepcionista
password: 12345
En caso de que el archivo se haya eliminado, se generará un usuario automáticamente

El técnico se encuentra guardado dentro de:
.../FrmPrincipal/bin/Debug/technical.xml
De todas formas los valores por defecto son:
username: tecnico
password: 12345
En caso de que el archivo se haya eliminado, se generará un usuario automáticamente

Implementación de las clases:

Excepciones: Dentro del proyecto “Exceptions” se encuentran algunas excepciones personalizadas, como por ejemplo “AlreadyInListException” que se lanza cuando una computadora ya se encuentra cargada en el sistema.

Está implementada en Entities.Computers -> Computer -> Sobrecarga ==

En Files -> TextHandler y FilesHandler -> ReadFile se implementa la excepción FileNotFoundException

Test unitarios: Dentro del proyecto “Test” existen 5 test unitarios que prueban las siguientes funcionalidades:

1. Agregar computadora
2. Eliminar computadora
3. Excepción de computadora repetida
4. Guardar en xml y leer

Tipos genéricos: Fue implementado dentro del proyecto “Files” tanto en las clases “FilesHandler”, “TextHandler” como en su respectiva interfaz “IFiles”.

Interfaces: Fue implementado en:

1. Files -> Interfaz IFiles
2. Entities.Users -> IUser, ITechnician, IReceptionist
3. Entities.Computers -> IComputer, INotebook, IDesktop
4. Procedure -> IProcedure

5. SignIn -> ISignIn

Archivos y serialización:

El código de archivos se encuentra en Files -> TextHandler y fue utilizado en el botón “Ayuda” leyendo un archivo de texto “help.txt”

FrmPrincipal -> FrmSignIn -> btnHelp_Click

Y al momento de manipular una computadora (Ya sea porque se cargó, se modificó, se reparó o se trasladó entre usuarios) se guarda un registro de los movimientos en “list_computers.txt”.

Procedure -> CoreProcedure -> AddComputer

Ambos archivos se encuentran en “.../FrmPrincipal/bin/Debug/”

El código de serialización se encuentra en Files -> File Handler y fue utilizado tanto para serializar como para deserializar User y Computers

Serialización de User: SignIn -> SignInHandler -> SaveFile

Deserialización de User: SignIn -> SignInHandler -> ReadFile

Serialización de Computer: Procedure -> CoreProcedure -> AddComputer

Deserialización de Computer: Procedure -> CoreProcedure -> LoadComputers

Los XML de los usuarios se encuentran en la ruta mencionada anteriormente y las computadoras en “Computers.xml”

Bases de datos:

El DAO de bases de datos se encuentra en el proyecto SQL y su implementación dentro del proyecto Procedure -> CoreProcedure tanto en el agregar, como en el modificar y eliminar computadoras

Delegados y eventos:

Se encuentran implementados en el formulario de inicio de sesión (FrmSignIn) en donde se utiliza un evento con un manejador de eventos suscrito para iniciar sesión ya sea cuando el botón fue presionado o porque se presionó ENTER.

También se encuentran implementados dentro de la clase CoreProcedure al momento de agregar, modificar o eliminar computadoras.

Metodos de extension:

El código se encuentra en Entities.Computers -> StateExtension y esta implementado en Entities.Computers.Desktop -> Desktop y

Entities.Computers.Notebook -> Notebook, el método retorna un string con el estado de la computadora separado en dos palabras (en caso que lo sea)

Threads:

Está implementado en FrmComputer, a la hora de reparar una computadora se completa un label de “Reparaste la computadora, ¡excelente!”.

La llamada a los eventos de este hilo están dentro de Procedure -> CoreProcedure

-> Update Computer, cada 100 ms agrega una nueva letra a la oración