ALCANCES EN LA FUNCIONALIDAD

|  |  |
| --- | --- |
| Acceso al sistema | * Se logro realizar un menú de login que permite al usuario acceder a las demás funciones del programa únicamente si existe el usuario que ingreso y si la contraseña es correcta. * Se logro realizar un menú de registro en el cual se puede crear nuevos usuarios con todos los datos propuestos en el enunciado. * En el menú de registro también se logró implementar la funcionalidad de la contraseña, es decir, permite registrarse únicamente si cumple con los requerimientos de seguridad vistos en los laboratorios en clase. * Se implemento que por defecto el primer usuario ingresado es administrador. * Se logro implementar que sólo si se ingresan todos los campos con corrección y se graba exitosamente en el archivo, se permite el ingreso a la aplicación |
| Menú de administración | * Se logro implementar los permisos de administrador y de usuarios, permitiendo a cada rol realizar únicamente las tareas que indica el enunciado. |
| Mantenimiento de Usuarios | Si logra realizar los cambios en los archivos cuando se modifica algún usuario y se mantienen los permisos de administrador y de usuarios. |
| Búsqueda de Usuarios | Permite ir a buscar a los archivos el usuario ingresado mostrando sus datos si la búsqueda se realiza como administrador o mostrando con un mensaje si existe o no el usuario si la búsqueda se realiza como usuario |
| Opción Respaldo de la Información | Permite copiar todos los archivos de la carpeta MEIA a una carpeta definida por el usuario. |
|  | |

**Estructuras, Clases y Métodos**

**CLASES:**

**BitacoraBackup:**

Atributos:

* private String rutaAbsoluta;
* private String usuario;
* private String fecha;

Métodos:

* public BitacoraBackup(String rutaAbsoluta, String usuario, String fecha)
* public String getRutaAbsoluta()
* public void setRutaAbsoluta(String rutaAbsoluta)
* public void setUsuario(String usuario)
* public String getFecha()
* public void setFecha(String fecha)

**DescriptorBitacoraBackup**

Atributos:

* String nombre=" ",fechaCreacion=" ",usuarioCreacion=" ",fechaModificacion=" ",usuarioModificacion=" ";
* int numRegistros=0;

Métodos:

* public String toString()
* public String getNombre()
* public void setNombre(String nombre)
* public String getFechaCreacion()
* public void setFechaCreacion(String fechaCreacion)
* public String getUsuarioCreacion()
* public void setUsuarioCreacion(String usuarioCreacion)
* public String getFechaModificacion()
* public void setFechaModificacion(String fechaModificacion)
* public String getUsuarioModificacion()
* public void setUsuarioModificacion(String usuarioModificacion)
* public int getNumRegistros()
* public void setNumRegistros(int numRegistros)

**DescriptorUsuarios**

Atributos:

* String nombre, fechaCreacion, usuarioCreacion, fechaModificacion, usuarioModificacion;

Métodos:

* public DescriptorUsuario(String nombre, String fechaCreacion, String usuarioCreacion, String fechaModificacion, String usuarioModificacion, int numRegistros, int registrosActivos, int registrosInactivos, int maxReorganizacion)
* public String getNombre()
* public void setNombre(String nombre)
* public String getFechaCreacion()
* public void setFechaCreacion(String fechaCreacion)
* public String getUsuarioCreacion()
* public void setUsuarioCreacion(String usuarioCreacion)
* public String getFechaModificacion()
* public void setFechaModificacion(String fechaModificacion)
* public String getUsuarioModificacion()
* public void setUsuarioModificacion(String usuarioModificacion)
* public int getNumRegistros()
* public void setNumRegistros(int numRegistros)
* public int getRegistrosActivos()
* public void setRegistrosActivos(int registrosActivos)
* public int getRegistrosInactivos()
* public void setRegistrosInactivos(int registrosInactivos)
* public int getMaxReorganizacion()
* public void setMaxReorganizacion(int maxReorganizacion)

**Escritor**

Métodos:

* public static boolean Escribir(String strPath,String strContenido)

**Imagen**

Métodos:

* public Imagen()
* public void paint(Graphics grafico)

**Lector**

Métodos:

* public static String Obtener (String strPath)

**LimitarCaracteres**

Atributos:

* private int limite;

Métodos:

* public LimitarCaracteres(int i)
* public void insertString(int compensacion, String cadena, AttributeSet attributeset)throws BadLocationException

**LimitarNumeros**

Métodos:

* public void insertString(int arg0, String arg1, AttributeSet arg2) throws BadLocationException

**OperacionesSecuencial**

Métodos:

* public static LinkedList<Usuario> obtenerUsuarios(int OpcionObtener)
* public static void rellenarUsuarios(LinkedList<Usuario> usuarios)
* public static void rellenarUsuariosMaestro(LinkedList<Usuario> usuarios)
* public static DescriptorUsuario obtenerDescriptorUsuario(int OpcionObtener)
* public static void rellenarDescriptorUsuario(DescriptorUsuario descriptor,int OpcionObtener)
* public static DescriptorBitacoraBackup obtenerDescriptorBB()
* public static void LlenarUsuariosMaestro() throws IOException
* public static int contarActivos(LinkedList<Usuario> lista,int retorno)
* public static String getMD5(String input)

**Usuario**

Atributos:

* private String usuario,nombre,apellido,contraseña,correo,

pathFoto,fechaNacimiento;

* private int rol,telefono,estatus, FixedSize;

Métodos:

* public static int getFixedSize()
* public void setFixedSize(int FixedSize)
* public String toString()
* public String toFixedSizeString()
* public Usuario(String usuario, String nombre, String apellido, String contraseña, String fechaNacimiento,String correo, String pathFoto, int telefono, int estatus)
* public String getUsuario()
* public void setUsuario(String usuario)
* public String getNombre()
* public void setNombre(String nombre)
* public String getApellido()
* public void setApellido(String apellido)
* public String getContraseña()
* public void setContraseña(String contraseña)
* public String getCorreo()
* public void setCorreo(String correo)
* public String getPathFoto()
* public void setPathFoto(String pathFoto)
* public int getRol()
* public void setRol(int rol)
* public int getTelefono()
* public void setTelefono(int telefono)
* public int getEstatus()
* public void setEstatus(int estatus)
* public String getFechaNacimiento()
* public void setFechaNacimiento(String fecha)
* public int compare(Usuario o1, Usuario o2)

**Cargar**

Métodos:

* public cargar(JProgressBar progreso)
* public void pausa(int seg)

**ProyectoMEIA**

Atributos:

* public static Usuario usuarioEnUso;

**INTERFASE:**

**Login**

Métodos:

public LogIn() throws IOException

**OpcionesAdministrador**

Métodos:

* public OpcionesAdministrador()
* private void copiarDirectorio(String origen, String destino)
* private void copiarArchivo(String sOrigen, String sDestino)
* private void comprobarCrearDirectorio(String ruta)
* public void AgregarBitacora(BitacoraBackup datosBitacora)

**OpcionesUsuario**

Métodos:

* public OpcionesUsuario()
* public void insertUpdate(DocumentEvent e)
* public int verificarContraseña(String texto)
* public int contarCoincidencias(String texto1,String texto2)

**RegistroUsuario**

Métodos:

* void limpiarCampos()
* boolean VerSiExisteUsuario(Usuario us)
* public void insertUpdate(DocumentEvent e)
* public int verificarContraseña(String texto)
* public int contarCoincidencias(String texto1,String texto2)
* public void removeUpdate(DocumentEvent e)
* public void changedUpdate(DocumentEvent e)
* public boolean VerificarCampos()

**PantallaCarga**

Métodos:

* public PantallaCarga()
* private void jProgressBar1StateChanged (javax.swing.event.ChangeEvent evt)

**BuscarUsuarios**

Atributos:

* boolean activoAntes=false,activoDespues=false;
* boolean adminAntes=true,adminDespues=true;
* boolean esAdmin=false;
* Usuario usuarioBuscado;

Métodos:

* public BuscarUsuarios()