**Prueba de Modelo de Calidad**

1. Funcionalidad

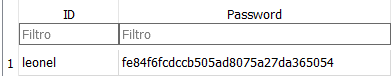
* **Seguridad de acceso**

Descripción:

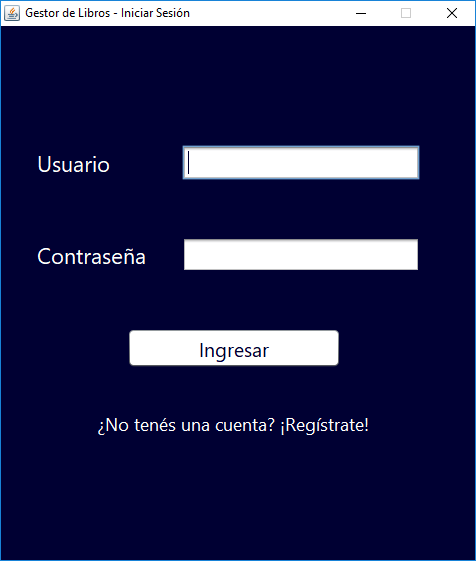
Capacidad del producto software para asegurar la integridad de los datos y la confidencialidad de estos.

**Características a medir:**

* **Encriptación de la contraseña y datos sensibles.**



* **Inicio de sesión.**

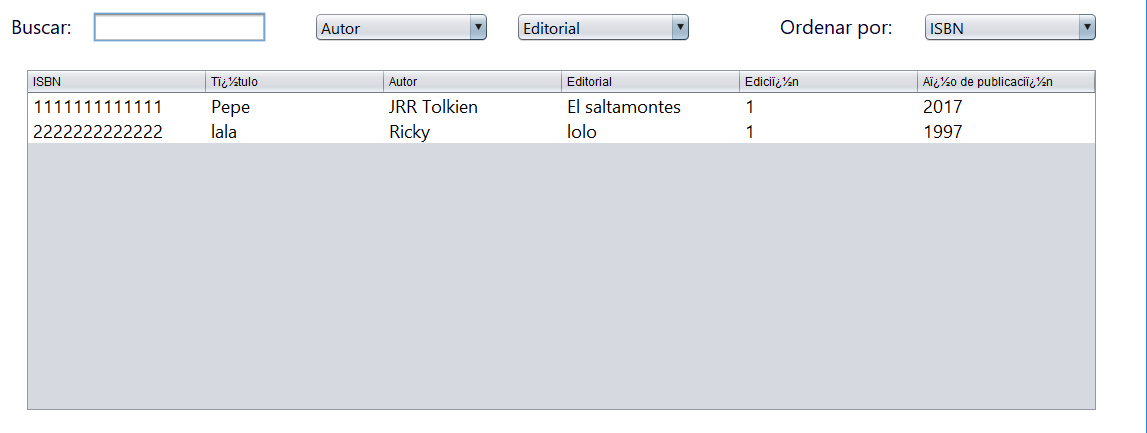


Evaluación: Al contar con ambas características, la **seguridad de acceso** es aceptable y suma 4 puntos.

* **Exactitud**

Descripción:

Es la capacidad del producto software para proporcionar los resultados con el grado necesario de precisión.





Evaluación: Al realizarse 3 informes, la **exactitud** es aceptable y suma 2 puntos.

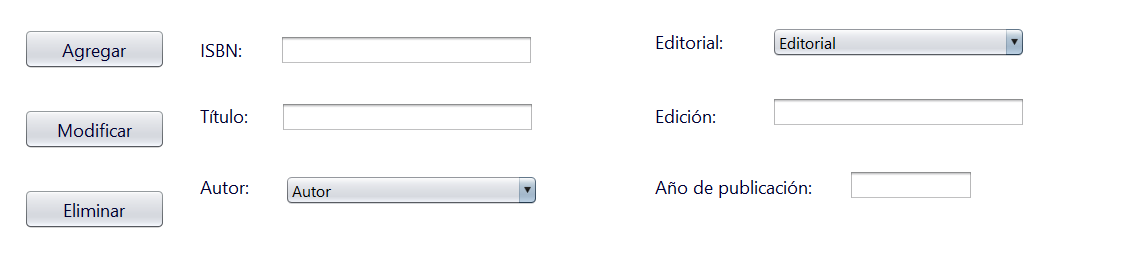
* **Adecuación**

Descripción:

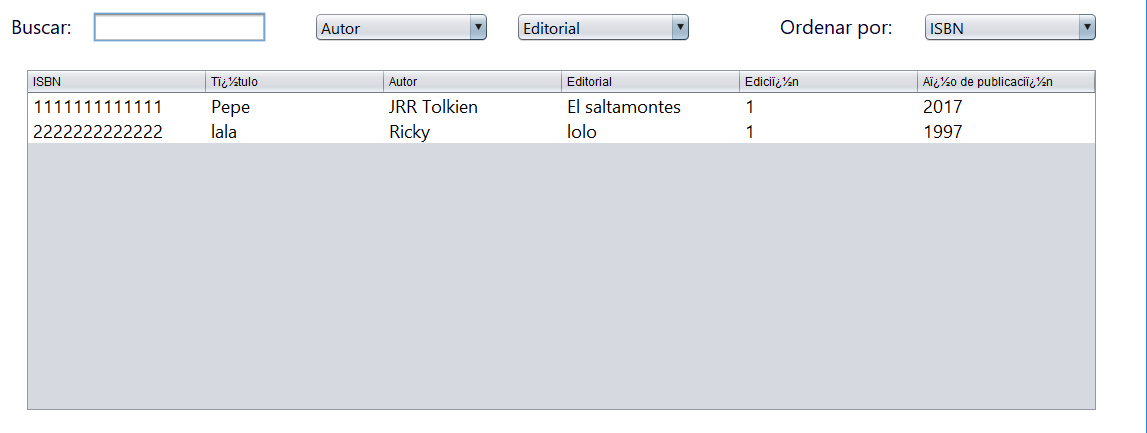
Capacidad del producto software para proporcionar un conjunto apropiado de

funciones para tareas y objetivos de usuario especificados

Insertar nuevo libro



Buscar libros



Evaluación: Al cumplir con al menos 2 funcionalidades requeridas, la **adecuación** es aceptable y suma 2 puntos.

**Subcaracterística Ponderada:** Seguridad de Acceso.

**Subtotal: 8 pts.**

2. Eficiencia

* **Utilización de los Recursos**

Descripción:

Se evaluará la eficiencia del producto software de acuerdo al porcentaje de uso de procesador y de memoria que realice.

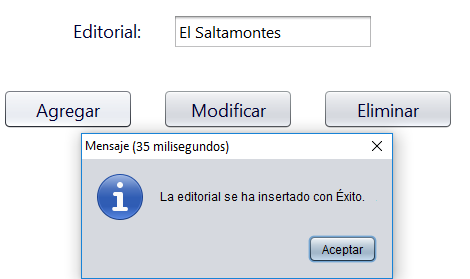
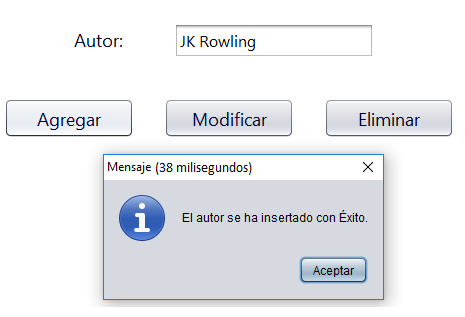


Evaluación: Al consumir menos del 25% de procesador y memoria, la **utilización de los recursos** es aceptable y suma 4 puntos.

* **Comportamiento temporal**

Descripción:

Se evaluará el tiempo que está el producto software sin informarle al usuario del estado en que se encuentra la solicitud que realizó.



Evaluación: Al demorarse menos de 2 segundos en informar el estado de la solicitud, el **comportamiento temporal** es aceptable y suma 2 puntos.

**Subcaracteristica Ponderada:** Utilización de los recursos.

**Subtotal: 6 pts.**

3. Fiabilidad

* **Tolerancia a fallos**

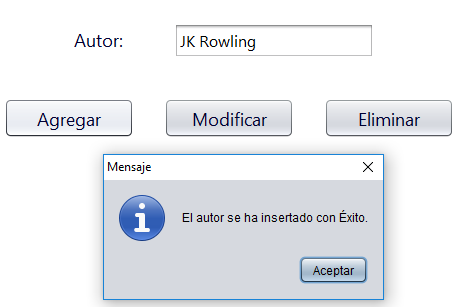
Descripción:

Es la capacidad del producto software de mantener la integridad de los datos cuando se producen fallas del sistema.

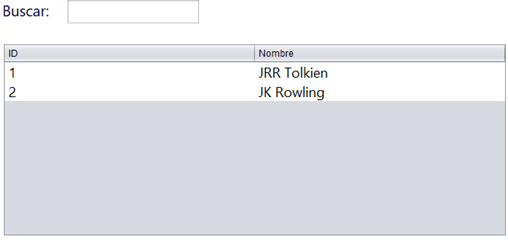
**Características a medir:**

**• Cuando sucede un error se protegen los datos procesados.**

Damos de alta a un autor



Procedemos a realizar el sistema, sin cerrar correctamente el programa. Una vez iniciado el sistema nuevamente, abrimos el programa y notamos que los datos ingresados se encuentran correctamente en la grilla de autores.



**• Se realiza un log de actividades que el sistema estaba haciendo.**

****

Evaluación: Al cumplir con las dos características, la **tolerancia a fallos** es aceptable y suma 4 puntos.

* **Capacidad de Recuperación de Errores**

Descripción:

Es la capacidad del sistema de reanudar sus actividades cuando se producen errores críticos.

Está característica no se ha contemplado.

**Características a medir:**

**• El sistema reanuda las actividades si se produce una falla crítica.**

**• Reanuda sus actividades y vuelve al estado en que estaba.**

Evaluación: Al no cumplir con ninguna característica, la **capacidad de recuperación de errores** es no aceptable y no suma puntos.

**Subcaracteristica Ponderada:** Tolerancia a fallos.

**Subtotal: 4 pts.**

4. Mantenibilidad

* **Capacidad de ser analizado:**

Descripción

Para evaluar la capacidad que tiene el código para ser analizado se tiene en cuenta el promedio de comentarios de todos los métodos del programa.

**Clase BD:**

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Cantidad de líneas | Líneas comentadas | % Líneas comentadas |
| BD() | 36 | 22 | 61,11 |
| deleteLibro() | 10 | 2 | 20 |
| getLibro() | 9 | 4 | 44,44 |
| getAll() | 12 | 5 | 41,67 |
| searchLibro() | 23 | 12 | 52,17 |
| insertAutor() | 10 | 2 | 20 |
| updateAutor() | 12 | 3 | 25 |
| deleteAutor() | 10 | 2 | 20 |
| getAutores() | 8 | 4 | 50 |
| searchAutor() | 12 | 6 | 50 |
| insertEditorial() | 10 | 2 | 20 |
| updateEditorial() | 12 | 3 | 25 |
| deleteEditorial() | 10 | 2 | 20 |
| getEditoriales() | 8 | 4 | 50 |
| searchEditorial() | 12 | 5 | 41,67 |
| insertUsuario() | 3 | 3 | 100 |
| deleteUsuario() | 2 | 2 | 100 |
| verificarLogin() | 9 | 3 | 33,33 |
| ordenarResultado() | 12 | 6 | 50 |
| close() | 20 | 20 | 100 |
| agregarTextoAlfinal() | 8 | 0 | 0 |

**Clase Entidad:**

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Cantidad de líneas | Líneas comentadas | % Líneas comentadas |
| Entidad() | 2 | 0 | 0 |
| getId() | 1 | 1 | 100 |
| setId() | 1 | 1 | 100 |
| getNombre() | 1 | 1 | 100 |
| setNombre() | 1 | 1 | 100 |

**Clase Gestor:**

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Cantidad de líneas | Líneas comentadas | % Líneas comentadas |
| Gestor() | 50 | 10 | 20 |
| windowClosing() | 11 | 3 | 27,27 |
| agregarFilaLibro() | 9 | 2 | 22,22 |
| llenarCamposLibro() | 8 | 1 | 12,5 |
| recuperarInformacionLibro() | 11 | 3 | 27,27 |
| actualizarTablaLibros() | 2 | 2 | 100 |
| limpiarTablaLibros() | 4 | 2 | 50 |
| limpiarCamposLibro() | 6 | 0 | 0 |
| buscarLibro() | 22 | 10 | 45,45 |
| traerTodosLosLibros() | 12 | 4 | 33,33 |
| insertarLibro() | 16 | 8 | 50 |
| modificarLibro() | 10 | 5 | 50 |
| eliminarLibro() | 8 | 3 | 37,5 |
| validarCamposLibro() | 18 | 4 | 22,22 |
| agregarFilaAutor() | 5 | 2 | 40 |
| actualizarTablaAutores() | 2 | 2 | 100 |
| limpiarTablaAutores() | 5 | 3 | 60 |
| buscarAutor() | 11 | 5 | 45,45 |
| insertarAutor() | 18 | 8 | 44,44 |
| modificarAutor() | 11 | 6 | 54,55 |
| eliminarAutor() | 8 | 4 | 50 |
| agregarFilaEditorial() | 5 | 2 | 40 |
| actualizarTablaEditoriales() | 2 | 2 | 100 |
| limpiarTablaEditoriales() | 5 | 3 | 60 |
| buscarEditorial() | 11 | 5 | 45,45 |
| insertarEditorial() | 19 | 8 | 42,11 |
| modificarEditorial() | 11 | 6 | 54,55 |
| eliminarEditorial() | 8 | 4 | 50 |
| cargarMenu() | 54 | 9 | 16,67 |
| cargarInicio() | 205 | 25 | 12,2 |
| isCellEditable() | 1 | 0 | 0 |
| cargarAdministracion() | 172 | 21 | 12,21 |
| cargarListeners() | 193 | 58 | 30,05 |
| actionPerformed() | 1 | 1 | 100 |
| keyReleased() | 1 | 1 | 100 |
| mouseClicked() | 4 | 3 | 75 |
| mouseReleased() | 9 | 2 | 22,22 |
| cargarAutoresYEditoriales() | 33 | 15 | 45,45 |
| informar() | 1 | 1 | 100 |

**Clase Launcher:**

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Cantidad de líneas | Líneas comentadas | % Líneas comentadas |
| main() | 20 | 3 | 15 |
| run() | 16 | 2 | 12,5 |

**Clase Libro:**

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Cantidad de líneas | Líneas comentadas | % Líneas comentadas |
| Libro() | 6 | 0 | 0 |
| equals() | 1 | 1 | 100 |
| compareTo() | 1 | 1 | 100 |
| toString() | 1 | 0 | 0 |
| getISBN() | 1 | 1 | 100 |
| setISBN() | 1 | 1 | 100 |
| getTitulo() | 1 | 1 | 100 |
| setTitulo() | 1 | 1 | 100 |
| getAutor() | 1 | 1 | 100 |
| setAutor() | 1 | 1 | 100 |
| getEditorial() | 1 | 1 | 100 |
| setEditorial() | 1 | 1 | 100 |
| getEdicion() | 1 | 1 | 100 |
| setEdicion() | 1 | 1 | 100 |
| getAnioPublicacion() | 1 | 1 | 100 |
| setAnioPublicacion() | 1 | 1 | 100 |

**Clase Login:**

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Cantidad de líneas | Líneas comentadas | % Líneas comentadas |
| Login() | 111 | 15 | 13,51 |
| actionPerformed() | 6 | 0 | 0 |
| mouseEntered() | 2 | 0 | 0 |
| mouseReleased() | 3 | 0 | 0 |
| login() | 12 | 8 | 66,67 |

**Clase Registro:**

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Cantidad de líneas | Líneas comentadas | % Líneas comentadas |
| main() | 21 | 0 | 0 |
| run() | 17 | 0 | 0 |
| Registro() | 130 | 18 | 13,85 |
| actionPerformed() | 6 | 2 | 33,33 |
| mouseEntered() | 2 | 0 | 0 |
| mouseReleased() | 7 | 0 | 0 |
| registrarse() | 27 | 10 | 37,04 |
| registrarEnDB() | 1 | 1 | 100 |

Evaluación: Al poseer un promedio de comentarios de 50,15%, la **capacidad de ser analizado** es aceptable y suma 2 puntos.

* **Capacidad para ser modificado:**

Descripción:

Para evaluar la capacidad que tiene el código para ser cambiado se tomará en cuenta la complejidad ciclomática del método.

|  |  |
| --- | --- |
| searchLibro() |  |
| ordenarResultado() |  |
| registrarse() |  |
| cargarListeners() |  |
| validarCamposLibro() |  |
| buscarLibro() |  |

Para esto se analizó la Complejidad Ciclomática de los 50 métodos existentes en el software. Sólo se muestran los métodos más significativos del sistema.

Evaluación: Al existir métodos con complejidad ciclomática entre 11 y 20, la **capacidad para ser modificado** es medianamente aceptable y suma 1 punto.

* **Estabilidad:**

Descripción:

Para determinar la estabilidad del software se evalúa el promedio de fallas que presenta el producto por prueba.

Se han realizado las pruebas del sistema y no se ha presentado ninguna falla, véase Pruebas del Sistema.

Evaluación: Al no presentar ninguna falla en sus pruebas, la **estabilidad** es aceptable y suma 4 puntos.

**Subcaracteristica Ponderada:** Estabilidad.

**Subtotal: 7 pts.**

5.Usabilidad

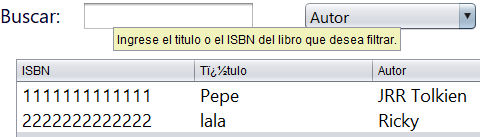
* **Capacidad de ser entendido**

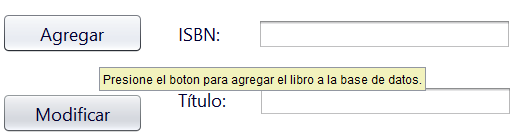
Descripción:

Capacidad que posee el software, para ayudar a los usuarios ante una determinada situación donde se necesite asistencia.

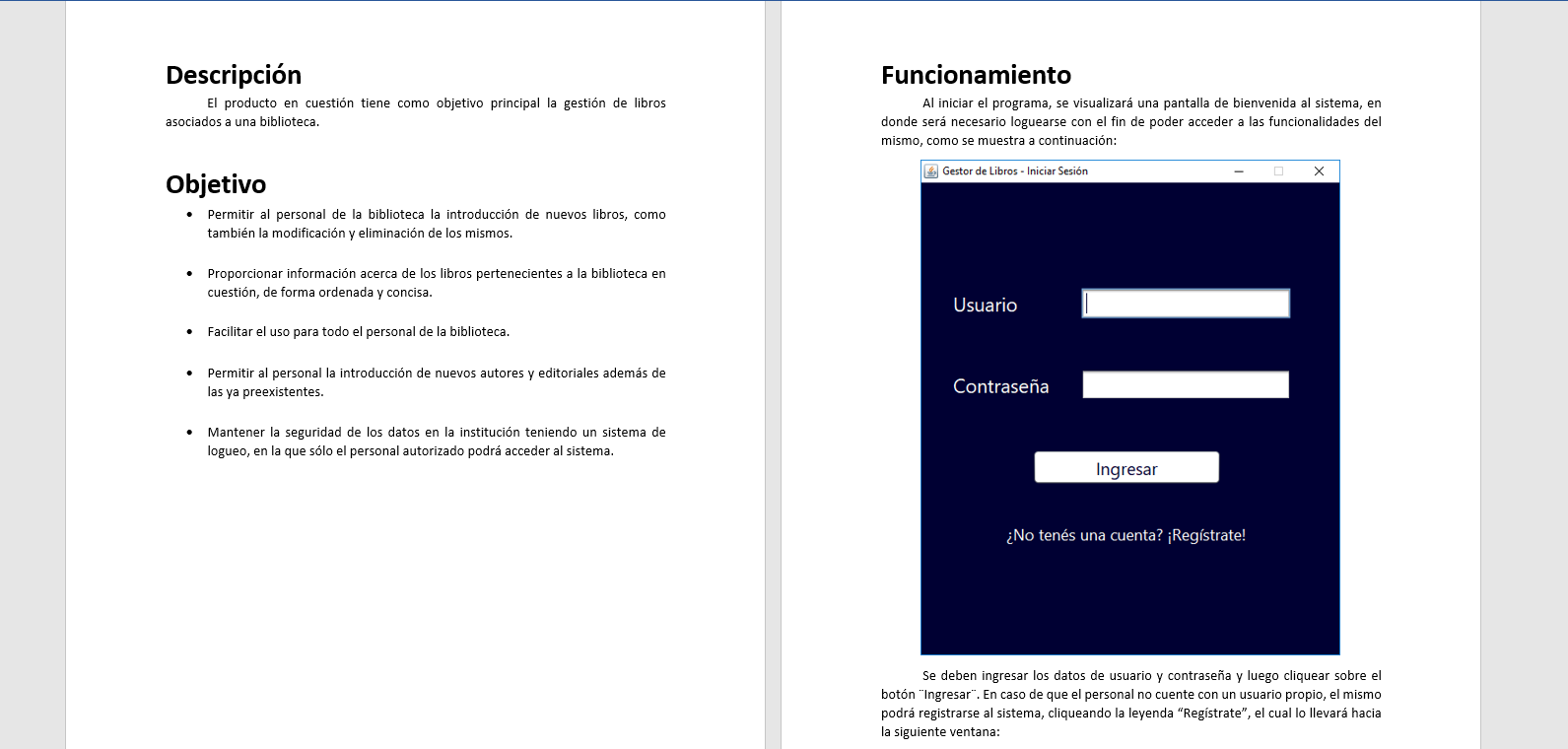
**Características a medir:**

**• Ayuda contextual sobre menús y botones de acción (tooltip).**

****

****

**• Manual de ayuda al usuario incorporado en el sistema.**

****

Evaluación: Al cumplir con las dos características, la **capacidad de ser entendido** es aceptable y suma 2 puntos.

* **Capacidad para ser operado**

Descripción

Es la capacidad del producto software de ser utilizado sin asistencia del manual de ayuda ni de un experto.

Para poder realizar esta prueba se le ha solicitado a un usuario sin conocimientos acerca del ámbito de IT que usara el software, comentándole brevemente que funciones podía realizar y aclarándole que disponía de un manual de usuario en caso de ser necesario. Luego de varios minutos investigando el software, el usuario no tuvo problemas para poder registrar libros, autores y editoriales como tampoco para realizar consultas.

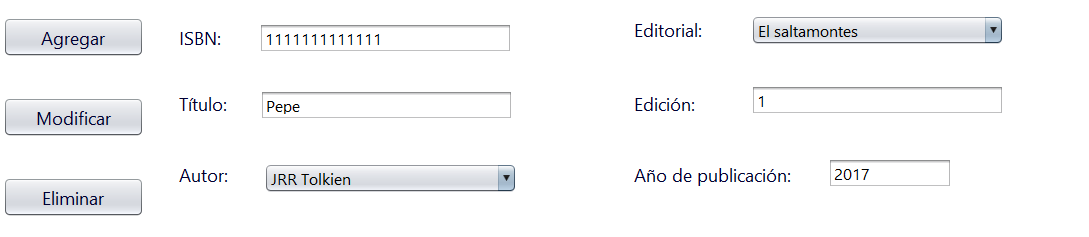
Evaluación: Al no requerir el manual de usuario para poder operar el software, la **capacidad para ser operado** es aceptable y suma 2 puntos

* **Capacidad de ser atractivo**

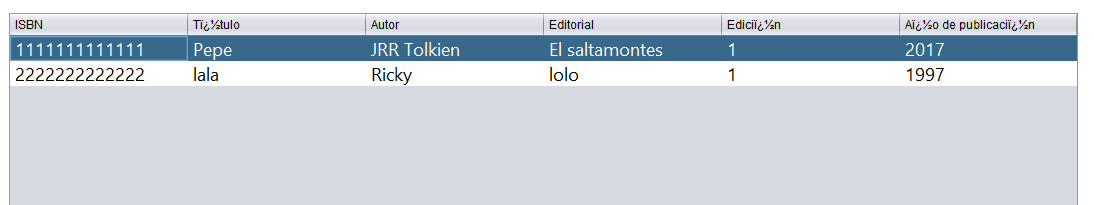
Descripción:

Es la agrupación correcta de funcionalidades dentro del programa en la interfaz gráfica para poder realizar una función específica.

Paso 1: Llenar datos del autor

****

Paso 2: Presionar Agregar y el libro se agregará con éxito



Evaluación: Al necesitar 3 o menos pasos promedio para ejecutar una funcionalidad, la **capacidad de ser atractivo** es aceptable y suma 4 puntos.

**Subcaracteristica Ponderada:** Capacidad de ser atractivo.

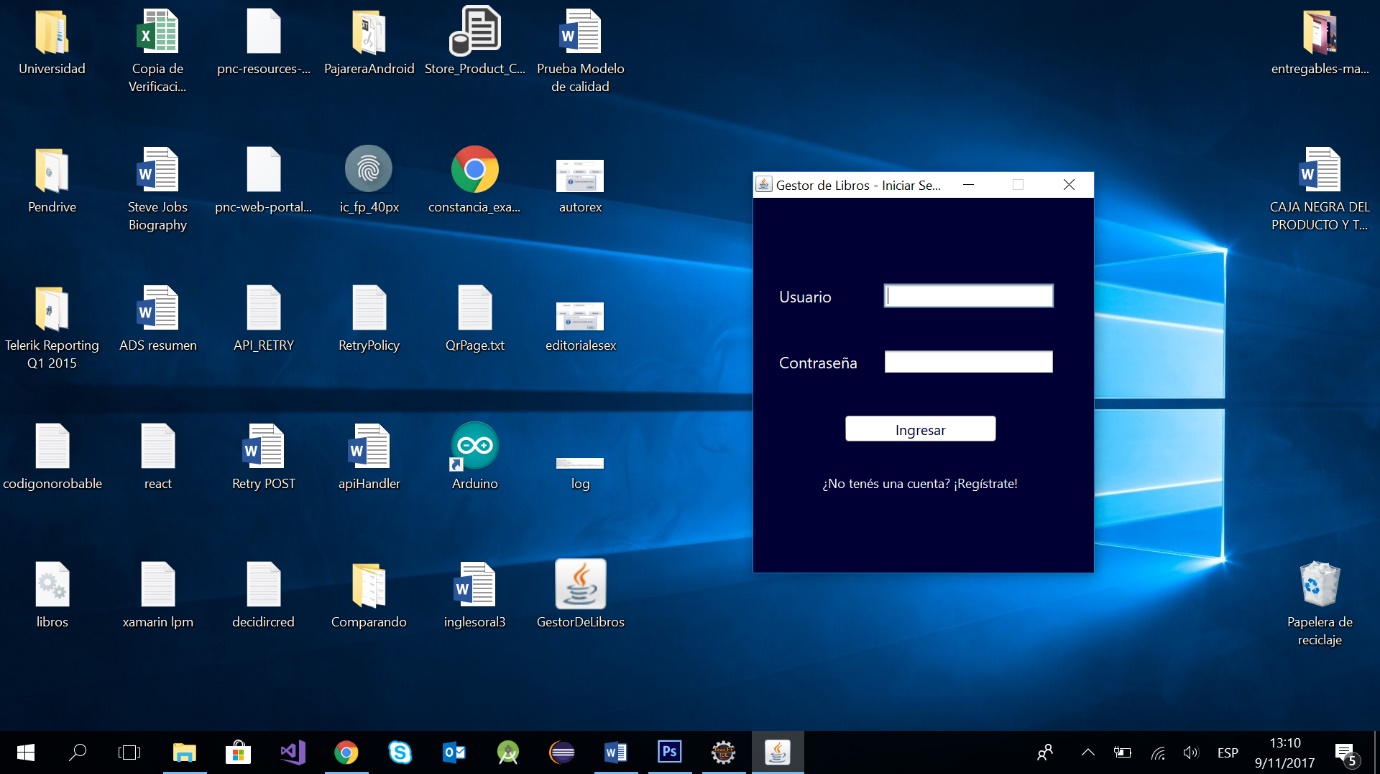
**Subtotal: 8 pts.**

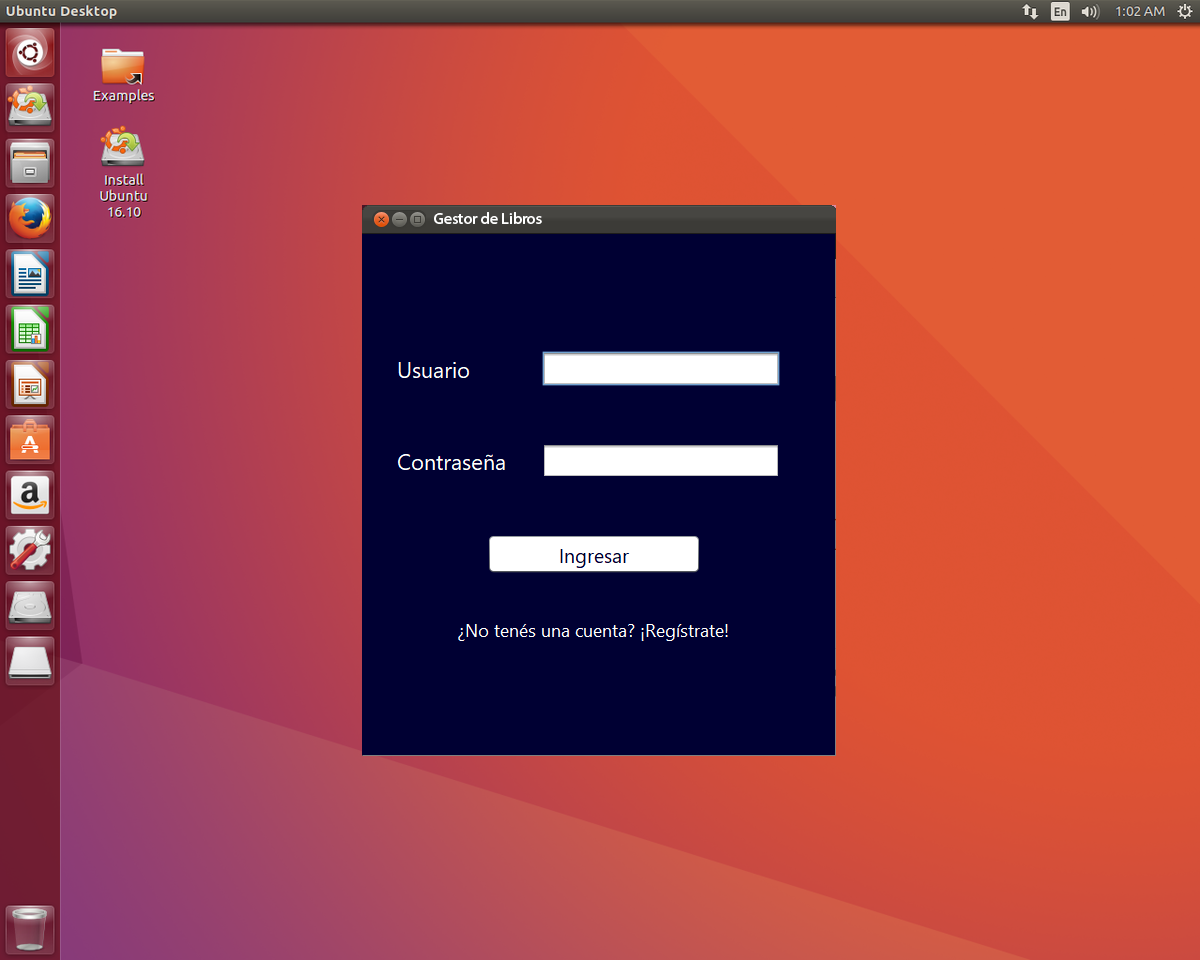
6. Portabilidad

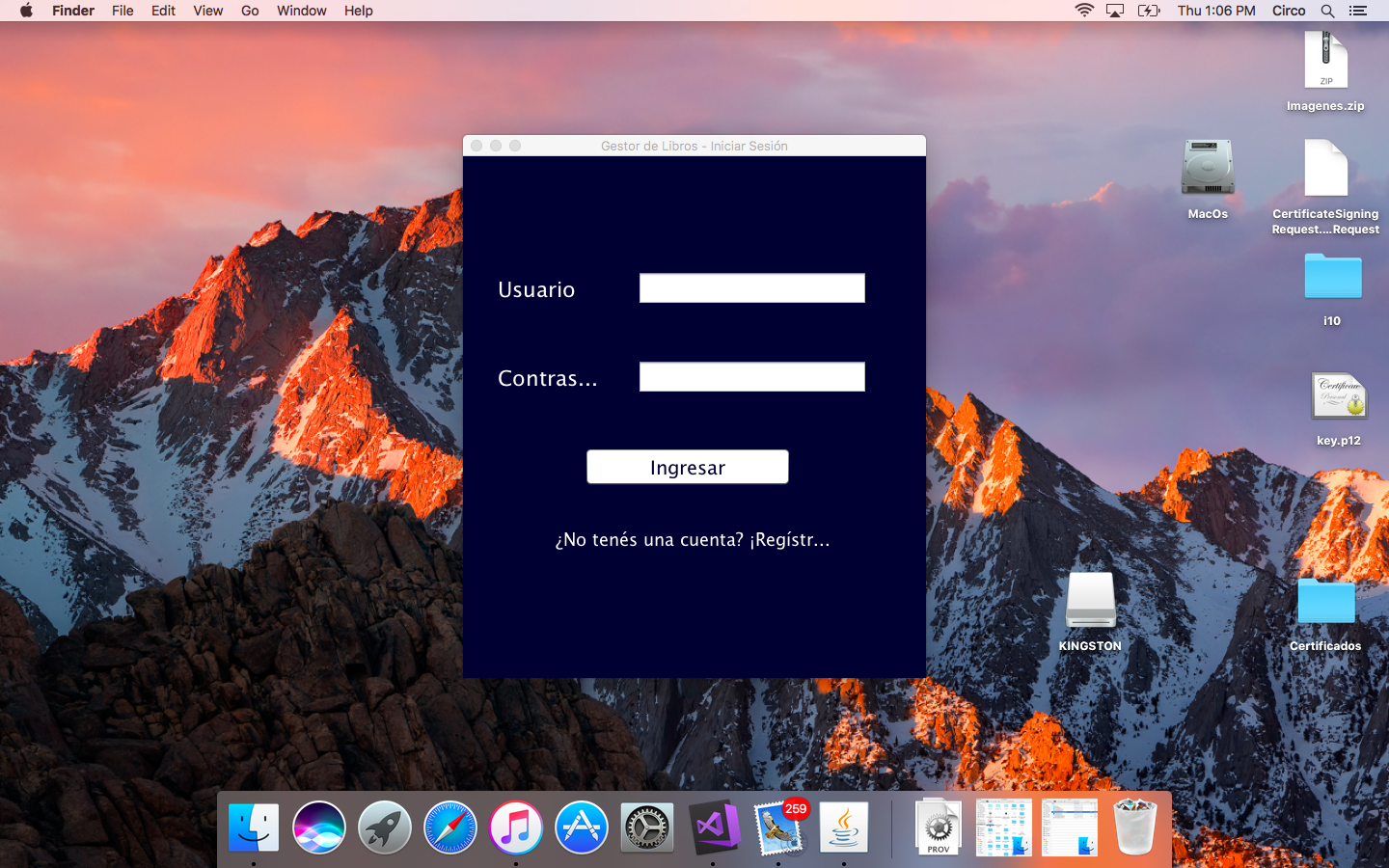
* **Adaptabilidad**

Descripción:

Es la capacidad del producto software de adaptarse a diferentes sistemas operativos sin cambiar su estructura interna.





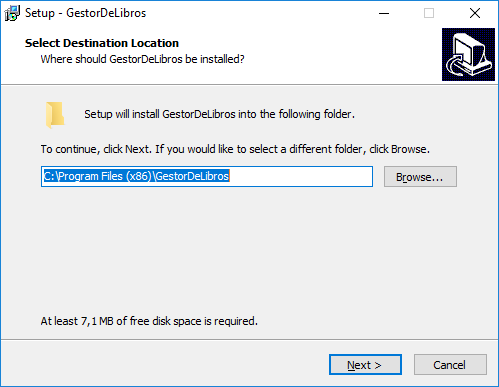
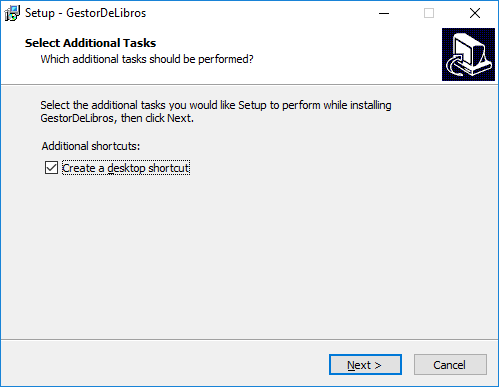


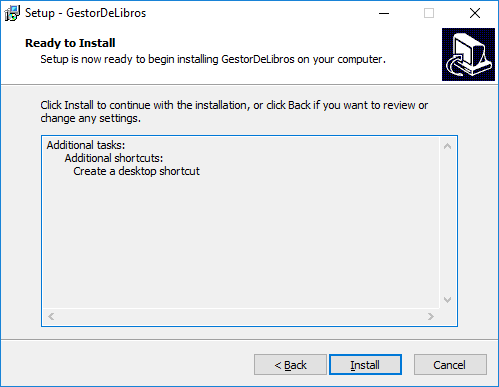
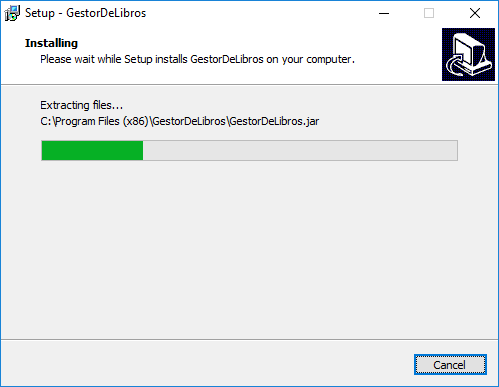
Evaluación: Al ser compatible con 3 o más sistemas operativos, la **adaptabilidad** es aceptable y suma 4 puntos.

* **Instalabilidad**

Descripción

El producto software debe poder ser instalado en una cantidad mínima de pasos.

Evaluación: Al instalarse en 4 pasos o menos, la **instalabilidad** es aceptable y suma 2 puntos.

**Subcaracteristica Ponderada:** Adaptabilidad.

**Subtotal: 6 pts.**

Finalmente se ejecuta el algoritmo de calidad con los datos tomados, y obtenemos que nuestro software cumple con el modelo de calidad, sumando un puntaje total de 39 puntos sobre 42 posibles. Y al tener cada característica su subcaracterística ponderada en aceptable, el resultado final del algoritmo es satisfactorio.

