



Universidad Nacional de La Matanza

Departamento de Ingeniería e
Investigaciones Tecnológicas

Análisis de Software

Trabajo Práctico

“Centro Médico California”

Grupo 7

Integrantes:

- Mediotte, Facundo
- Perez, Lautaro
- Amoruso, Sergio
- Reynoso, Thomas
- Brude, Alejandro
- Siculin, Luciano
- Monardo, Javier
- Gonzalez, Pablo

Índice

Descripción de la Funcionalidad	3
Modelo de Calidad.....	4
Funcionalidad	4
Eficiencia	5
Fiabilidad	6
Mantenibilidad	7
Usabilidad.....	8
Portabilidad	10
Algoritmo de evaluación de Modelo de Calidad	11
Métodos de Caja Blanca	12
Método de McCabe.....	14
Métodos de Cobertura.....	15
Método de Caja Negra.....	16
Pruebas de Sistema	17
Manual de usuario.....	18
Encuesta de Satisfacción de Usuario y Comparación	23
Cubierta de CD (ISO 9127)	26

Descripción de la Funcionalidad

El programa de control de pacientes del centro médico “California” nos permite realizar varias acciones con respecto a nuestro pacientes y médicos:

Ingreso de Datos:

- Datos del Paciente: Nos permite registrar un paciente en una base de datos ingresando un código, su apellido y nombre.
- Datos del Medico: Nos permite registrar un médico en una base de datos ingresando un código, su apellido y nombre.
- Diagnostico Paciente: Ingresando el código del paciente y el médico, podremos ingresar un diagnostico que será guardado para luego realizar el informe.

Informes:

- Listado de Pacientes por médico: Nos mostrará, dependiendo del médico seleccionado, los pacientes que el mismo atiende.
- Médicos por enfermedad: Dependiendo de la especialidad que seleccionemos nos mostrará todos los médicos que tratan esa enfermedad.

Modelo de Calidad

Funcionalidad

A) 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 datos
- Inicio de sesión de usuarios

Evaluación

- Mala [0]: No cumple con alguna característica.
- Regular [1]: Cumple con 1 característica.
- Buena [2]: Cumple con 2 características.

RESULTADOS OBTENIDOS

Regular [1]:

- El software no cuenta con encriptación de datos en la Base de Datos.
- La aplicación posee un inicio de sesión de usuarios para poder ser utilizada.

B) EXACTITUD DE LOS RESULTADOS

Descripción

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

Evaluación

- Mala [$\geq 10^{-3}$]: Los resultados tienen un error del orden de 10^{-3} o superior.
- Regular [$10^{-4}; 10^{-6}$]: Los resultados tienen un error del orden entre 10^{-4} y 10^{-6} .
- Buena [$\leq 10^{-7}$]: Los resultados tienen un error del orden de 10^{-7} o inferior.

RESULTADOS OBTENIDOS

Buena [$\leq 10^{-7}$]

- Las consultas a la Base de Datos son precisas obteniendo los datos a los cuales se quiere acceder.

Eficiencia

A) UTILIZACIÓN DE RECURSOS

Descripción

Se evaluará la eficiencia del producto software de acuerdo con el porcentaje de uso de procesador que realice.

Evaluación

- Mala [41;100]: 41% o más de uso de procesador.
- Regular [11;40]: 11% a 40% de uso de procesador.
- Buena [0;10]: 10% o menos de uso de procesador.

RESULTADOS OBTENIDOS

Buena [0;10]:

- El procesador no muestra signos de actividad significativos al encontrarse el programa en ejecución.
- Tampoco demuestra alteraciones de consumo al realizar consultas y/o acciones de guardado.

B) COMPORTAMIENTO EN EL TIEMPO

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

- Mala [≥ 5]: El producto está 5 o más segundos sin informar al usuario del estado de la solicitud.
- Regular [2;4]: El producto está entre 2 y 4 segundos sin informar al usuario del estado de la solicitud.
- Buena [0;1]: El producto está menos de 1 segundo sin informar al usuario del estado de la solicitud.

RESULTADOS OBTENIDOS

Buena [0;1]:

- El tiempo de respuesta del software es menor al segundo, y se encuentra en el orden de los milisegundos.

Fiabilidad

A) TOLERANCIA A FALLOS

Descripción

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

Características que medir

- Cuando sucede un error se protegen los datos procesados.
- Se realiza un log de actividades que el sistema estaba haciendo.

Evaluación

- Mala [0]: No cumple con alguna característica.
- Regular [1]: Cumple con 1 característica.
- Buena [2]: Cumple con 2 características.

RESULTADOS OBTENIDOS

Regular [1]:

- Al producirse un error, como cierre inesperado del programa, los datos que fueron ingresados quedan confirmados en la Base de datos.
- El sistema no realiza un log de actividades.

B) CAPACIDAD DE RECUPERACIÓN DE ERRORES

Descripción

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

Características que medir

- El sistema reanuda las actividades si se produce una falla crítica.
- Reanuda sus actividades y vuelve al estado en que estaba.

Evaluación

- Mala [0]: No cumple con ninguna característica.
- Regular [1]: Cumple con 1 característica.
- Buena [2]: Cumple con 2 características.

RESULTADOS OBTENIDOS

Buena [2]:

- Al ocurrir un fallo al intentar guardar en la base de datos, el sistema sigue funcionando normalmente sin interrumpir la interacción con el usuario.
- El sistema sigue en proceso a pesar de que se genere una falla.

Mantenibilidad

A) CAPACIDAD DEL CODIGO PARA SER ANALIZADO

Descripción

Para evaluar la capacidad que tiene el código para ser analizado se tiene en cuenta el porcentaje de comentarios que posee el código por cada método y en general.

Evaluación

- Mala [0;14]: 14% o menos del código comentado.
- Regular [15;29]: Entre 15 y 29% del código comentado.
- Buena [≥ 30]: 30% o más del código comentado

RESULTADOS OBTENIDOS

Regular [15;29]:

- 25% del código se encuentra comentado.

B) CAPACIDAD DEL CODIGO PARA SER CAMBIADO

Descripción

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

Evaluación

- Mala [21]: La complejidad ciclomática es mayor o igual a 21.
- Regular [11;20]: La complejidad ciclomática es entre 11 y 20.
- Buena [1;10]: La complejidad ciclomática es menor o igual a 10.

RESULTADOS OBTENIDOS

Buena [8]: La mayor complejidad ciclomática que presenta un método del sistema es de 8.

C) ESTABILIDAD

Descripción

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

Evaluación

- Mala [5]: El software presenta un promedio 5 o más errores por prueba.
- Regular [2;4]: El software presenta un promedio entre 2 y 4 errores por prueba.
- Buena [0;1]: El software presenta un promedio entre 0 y 1 error por prueba.

RESULTADOS OBTENIDOS

Buena [0;1]:

- El software presenta entre 0 y 1 errores por cada prueba realizada.

Usabilidad

A) CAPACIDAD DE SER ENTENDIDO

Descripción

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

Características que medir

- Posee ayuda contextual sobre menús y botones de acción.
- Manual de usuario incorporado al sistema como un menú dedicado.

Evaluación

- Mala [0]: No cumple con alguna característica.
- Regular [1]: Cumple con 1 característica.
- Buena [2]: Cumple con 2 características.

RESULTADOS OBTENIDOS

Buena [2]:

- El software posee interfaz gráfica con ayuda contextual.
- El sistema posee un manual de usuario que indica cómo funciona el programa.

B) CAPACIDAD DE SER OPERADO

Descripción

Es la Capacidad del producto software de ser utilizado sin asistencia adicional. Se valúa qué requiere el usuario para operar correctamente el producto.

Evaluación

- Mala [1]: El usuario requiere consultar a personal especializado para operar el producto software.
- Regular [2]: El usuario requiere ayuda contextual y manual de uso para operar el producto software.
- Buena [3]: El usuario opera el producto software sin asistencia.

RESULTADOS OBTENIDOS

Regular [2]:

- El usuario puede hacer uso del software sin necesidad de consultar de personal especializado, aunque sería necesario algún tipo de ayuda contextual para poder operarlo correctamente.

C) CAPACIDAD DE SER ATRACTIVO PARA EL USUARIO

Descripción

Es la agrupación correcta de funcionalidad del producto software en su interfaz gráfica, desde su agrupación lógica hasta el número promedio de pasos para alcanzar una función o contenido específico.

Evaluación

- Mala [6]: 6 o más pasos promedio sin organización de categoría.
- Regular [3;5]: Entre 3 y 5 pasos promedio y distribuidos en categorías.
- Buena [1;2]: 1 o 2 pasos promedio y distribuidos en categorías.

RESULTADOS OBTENIDOS

Buena [1;2]:

- El software presenta una interfaz amigable al usuario, debido a que permite realizar las operaciones en una cantidad corta de pasos (entre 1 y 2), en donde se muestran distintos tipos de menú, según qué función se quiera realizar.

Portabilidad

A) ADAPTABILIDAD

Descripción

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

Evaluación

- Mala [1]: Compatible con 1 sistema operativo.
- Regular [2]: Compatible con 2 sistemas operativos.
- Buena [≥ 3]: Compatible con 3 o más sistemas operativos.

RESULTADOS OBTENIDOS

Regular [2]:

- Se probó el programa tanto en Linux como en Windows, y ha funcionado correctamente.

B) INSTALABILIDAD

Descripción

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

Evaluación

- Mala [>7]: El producto se instala en 7 o más pasos.
- Regular [4;6]: El producto se instala entre 4 y 6 pasos.
- Buena [1;3]: El producto se instala en 3 o menos pasos.

RESULTADOS OBTENIDOS

Buena [1;3]

- El producto se instala en 3 pasos.

Algoritmo de evaluación de Modelo de Calidad

- Mala: 0 puntos
- Regular: 3 puntos
- Buena: 5 puntos

El software aprueba el modelo de calidad con 42 puntos o más. Y no puede tener ninguna valoración mala.

Funcionalidad:

- | | |
|---------------------------------------|-------|
| A. Seguridad de acceso: Regular | 3pts. |
| B. Exactitud de los resultados: Buena | 5pts. |

Eficiencia:

- | | |
|---------------------------------------|-------|
| A. Utilización de recursos: Buena | 5pts. |
| B. Comportamiento en el tiempo: Buena | 5pts. |

Fiabilidad:

- | | |
|------------------------------------------------|-------|
| A. Tolerancia a fallos: Regular | 3pts. |
| B. Capacidad de recuperación de errores: Buena | 5pts. |

Mantenibilidad:

- | | |
|-----------------------------------------------------|-------|
| A. Capacidad del código para ser analizado: Regular | 3pts. |
| B. Capacidad del código para ser cambiado: Buena | 5pts. |
| C. Estabilidad: Buena | 5pts. |

Usabilidad:

- | | |
|------------------------------------------------------|-------|
| A. Capacidad de ser entendido: Buena | 5pts. |
| B. Capacidad de ser operado: Regular | 3pts. |
| C. Capacidad de ser atractivo para el usuario: Buena | 5pts. |

Portabilidad:

- | | |
|---------------------------|-------|
| A. Adaptabilidad: Regular | 3pts. |
| B. Instalabilidad: Buena | 5pts. |

Puntos totales obtenidos: 60 puntos.

El software aprueba el modelo de calidad

Métodos de Caja Blanca

-Método elegido: *jButtonIngresarActionPerformed* de la clase *Login*.

```
private void jButtonIngresarActionPerformed(java.awt.event.ActionEvent evt) {  
    //GEN-FIRST:event_jButtonIngresarActionPerformed  
    String usuario = jTextFieldUser.getText();  
    String pass = String.valueOf(jPasswordField1.getPassword()); } 1  
    //Valido que ningún campo este vacío  
    if(usuario != null && !usuario.isEmpty() && pass!=null && !pass.isEmpty()){ } 2,3,4,5  
        User user = new User(usuario, pass); } 6  
        try {  
            //obtengo la conexión  
            factory = SessionFactoryManager.getSessionFactory();  
            UserR userRepo = new UserR(factory);  
  
            //Busco el usuario en la base de datos  
            User retorno = (User) userRepo.getByld(User.class, user.getUsuario());  
            if(retorno != null){ } 8  
                //si existe comparo las contraseñas  
                if(user.getPassword().equals(retorno.getPassword())){ } 9  
                    //si las contraseñas son iguales le abro la ventana principal  
                    Programa ventana = new Programa();  
                    ventana.requestFocusInWindow();  
                    ventana.setVisible(true);  
                    this.setVisible(false);  
                }else{  
                    //si la contraseña no es igual le informo que existe un error en las credenciales  
                    jLabelRespuestaE.setText("Usuario y/o contraseña incorrectos"); } 11  
                }  
            }else{  
                //si el usuario no existe le informo un error de credenciales  
                jLabelRespuestaE.setText("Usuario y/o contraseña incorrectos"); } 12  
            }  
            userRepo.getFactory().close(); } 13  
        } catch (HibernateException e) {  
            System.out.println(e); } 14  
        } finally {  
            factory.close(); } 15  
        }  
    }else {  
        jLabelRespuestaE.setText("Complete ambos campos"); } 16  
    }  
} } 17  
//GEN-LAST:event_jButtonIngresarActionPerformed
```

Elección:

Elegimos este método, ya que si bien no posee un alto FAN IN (Cantidad de métodos que lo llaman), es un método bastante importante dentro de nuestro programa, y posee una complejidad ciclomática alta en comparación a los demás métodos.

Nuestra herramienta de testing nos mostró el siguiente reporte sobre el método elegido:

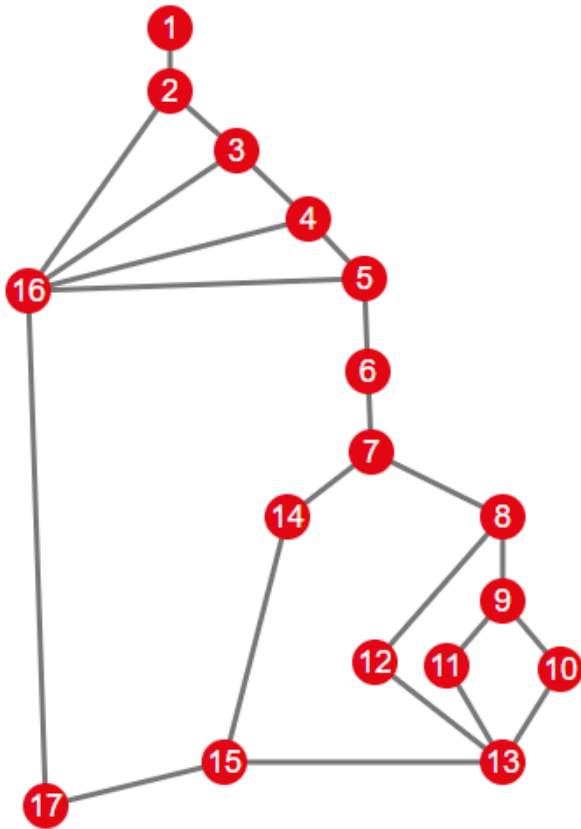
RESULTADOS DEL ANÁLISIS	
Líneas de código totales:	39
Líneas de código comentadas:	9
Porcentaje de comentarios:	23,08 %
Complejidad Ciclomática:	8
FAN-IN:	1
FAN-OUT:	15
HALSTEAD - Longitud:	65
HALSTEAD - Volumen:	327,89

Recomendación al tester:

Nuestras recomendaciones a tester que quiera probar nuestro producto son:

- Seleccionar los métodos que tengan el más alto FAN-IN, y por lo tanto serán los métodos más llamados dentro del programa, y realizarle las pruebas más exhaustivas. La prueba que se le podría realizar a estos métodos es la de complejidad ciclomática de McCabe. Lo ideal, sería poder lograr un TER lo más cercano a 100%.
- Los métodos que si bien no tiene un FAN-IN muy alto, pero son muy representativos para nuestro programa, probarlos con el método de cobertura de condición/decisión.
- Los demás métodos, si es que se quieren y pueden probar, realizarle el método de cobertura de sentencias.

Método de McCabe



Complejidad Ciclomática:

$V(G) = \text{Cantidad de Regiones} = 8$

$V(G) = \text{Nodos Predicados} + 1 = 7 + 1 = 8$

$V(G) = \text{Aristas} - \text{Nodos} + 2 = 23 - 17 + 2 = 8$

Complejidad Ciclomática: 8

Caminos:

- 1) 1 – 2 – 16 – 17
- 2) 1 – 2 – 3 – 16 – 17
- 3) 1 – 2 – 3 – 4 – 16 – 17
- 4) 1 – 2 – 3 – 4 – 5 – 16 – 17
- 5) 1 – 2 – 3 – 4 – 5 – 6 – 7 – 14 – 15 – 17
- 6) 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 12 – 13 – 15 – 17
- 7) 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 11 – 13 – 15 – 17
- 8) 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 13 – 15 – 17

Casos de Prueba:

N°	Usuario	Pass	Salida Esperada	Salida Obtenida
1	null	null	Complete ambos campos	Complete ambos campos
2		null	Complete ambos campos	Complete ambos campos
3	Lautaro	null	Complete ambos campos	Complete ambos campos
4	Lautaro		Complete ambos campos	Complete ambos campos
5	Lautaro	123	Error al traer de la Base de Datos	Error al traer de la Base de Datos
6	Lautaro	1234	Usuario y/o contraseña incorrectos	Usuario y/o contraseña incorrectos
7	Lautaro	12345	Usuario y/o contraseña incorrectos	Usuario y/o contraseña incorrectos
8	Lautaro	123456	Ingreso al Sistema	Ingreso al Sistema

Métodos de Cobertura

Condiciones y decisiones:

- C1: usuario != null
- C2: !usuario.isEmpty()
- C3: pass!=null
- C4: !pass.isEmpty()
- C5: User retorno = (User) userRepo.getByld(User.class, user.getUsuario());
- C6: retorno != null
- C7: user.getPassword().equals(retorno.getPassword())
- D1: (usuario != null && !usuario.isEmpty()) && pass!=null && !pass.isEmpty())
- D2: User retorno = (User) userRepo.getByld(User.class, user.getUsuario());
- D3: (retorno != null)
- D4: (user.getPassword().equals(retorno.getPassword()))

*Se considera la Condición 5 y la Decisión 2 por la presencia del try-catch.

-Método de condición:

Usuario	Pass	C1	C2	C3	C4	C5	C6	C7
NULL	NULL	F	-	-	-	-	-	-
	NULL	T	F	-	-	-	-	-
Lautaro	NULL	T	T	F	-	-	-	-
Lautaro		T	T	T	F	-	-	-
Lautaro	123	T	T	T	T	F		
Lautaro	1234	T	T	T	T	T	F	-
Lautaro	12345	T	T	T	T	T	T	F
Lautaro	123456	T	T	T	T	T	T	T

-Método de decisión:

Usuario	Pass	D1	D2	D3	D4
Lautaro		F	-	-	-
Lautaro	123	T	F	-	-
Lautaro	1234	T	T	F	-
Lautaro	12345	T	T	T	F
Lautaro	123456	T	T	T	T

-Método de condición/decisión:

Usuario	Pass	C1	C2	C3	C4	D1	C5	D2	C6	D3	C7	D4
NULL		F	-	-	-	F	-	-	-	-	-	-
		T	F	-	-	F	-	-	-	-	-	-
Lautaro	NULL	T	T	F	-	F	-	-	-	-	-	-
Lautaro		T	T	T	F	F	-	-	-	-	-	-
Lautaro	123	T	T	T	T	T	F	F	-	-	-	-
Lautaro	1234	T	T	T	T	T	T	T	F	F	-	-
Lautaro	12345	T	T	T	T	T	T	T	T	T	F	F
Lautaro	123456	T	T	T	T	T	T	T	T	T	T	T

Método de Caja Negra

-Método elegido: Ingresar de la clase Login de Usuarios:

Restricciones de las entradas:

- Usuario: No puede ser blanco ni null (cuando se presiona ingresar sin ingresar ningún valor)
- Contraseña: No puede ser blanco ni null (cuando se presiona ingresar sin ingresar ningún valor)

Son las únicas restricciones, ya que el encargado de crear la base de datos es Hibernate, por lo que no hay restricciones de longitud ni tipo de datos. Tampoco tenemos restricciones de valores, por lo que no podremos probar los valores límites.

Entrada	Condición	Clases Válidas	Clases no Válidas	Regla
Usuario	No puede ser blanco	i= [blanco] [1]	== [blanco] [2]	Booleana
	No puede ser null	i= [null] [3]	== [null] [4]	Booleana
Contraseña	No puede ser blanco	i= [blanco] [5]	== [blanco] [6]	Booleana
	No puede ser null	i= [null] [7]	== [null] [8]	Booleana

Casos de Prueba:

Caso		Clases cubiertas	Salida Esperada
Usuario	Contraseña		
Lautaro	123	1, 3, 5, 7	Ingreso al Sistema
	123	2	Complete ambos campos
null	123	4	Complete ambos campos
Lautaro		6	Complete ambos campos
Lautaro	null	8	Complete ambos campos

Pruebas de Sistema

Pruebas de Recuperación:

- Ambiente: Consisten en forzar el fallo del software y comprobar que la recuperación se lleva a cabo de manera correcta, devolviendo al sistema a un estado coherente.
- Precondiciones: El sistema debe estar funcionando y con datos cargados en la base de datos.
- Postcondiciones: El sistema deberá seguir mostrando los mismos datos que tenía anteriormente.

Hemos realizado la prueba generando un apagado repentino de la máquina donde estaba corriendo el sistema, y hemos obtenido los resultados esperados, ya que la base de datos estaba en el estado que tenía antes del apagón.

Pruebas de Seguridad:

- Ambiente: Intentan verificar que los mecanismos de protección incorporados al sistema lo protegerán, de hecho, de penetraciones inadecuadas.
- Precondiciones: Tenemos que tener el programa abierto y estar en la pantalla de Login. Se intentará ingresar al sistema con un usuario no registrado.
- Postcondiciones: El sistema deberá mostrar el mensaje Usuario y/o contraseña incorrectos.

Hemos realizado la prueba y hemos obtenido los resultados que esperábamos, ya que el programa cuenta con una validación de usuarios.

Pruebas de Resistencia:

- Ambiente: Estas pruebas están diseñadas para que el sistema requiera recursos en cantidad, frecuencia o volumen anormales. La idea es intentar que el sistema se venga abajo por la excesiva tensión a la que se lo somete.

No hemos podido realizar esta prueba, ya que no hay manera de hacer que el sistema requiera recursos en volúmenes, cantidad o frecuencia anormales. Nuestros clientes solo podrán ingresar de un valor a la vez, y el sistema toma los datos de una base de datos que tampoco se puede forzar.

Pruebas de Rendimiento:

- Ambiente: Es inaceptable que el software proporcione las funciones requeridas fuera de las condiciones de rendimiento exigidas.
- Precondiciones: Se debe tener el programa iniciado, y los datos de 50 pacientes, 50 médicos, y 100 diagnósticos. Se intentará ingresar toda esta información, y luego consultarla para verificar el correcto funcionamiento.
- Postcondiciones: El sistema debe seguir funcionando de la manera correcta, y debe haber guardado toda la información ingresada, sin tener ninguna fuga.

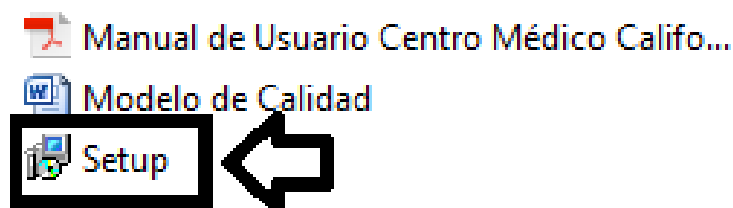
Hemos realizado la prueba y hemos obtenido los resultados esperados.

Manual de usuario

El presente es un instructivo para la correcta utilización del Sistema de control de médicos y pacientes del centro médico.

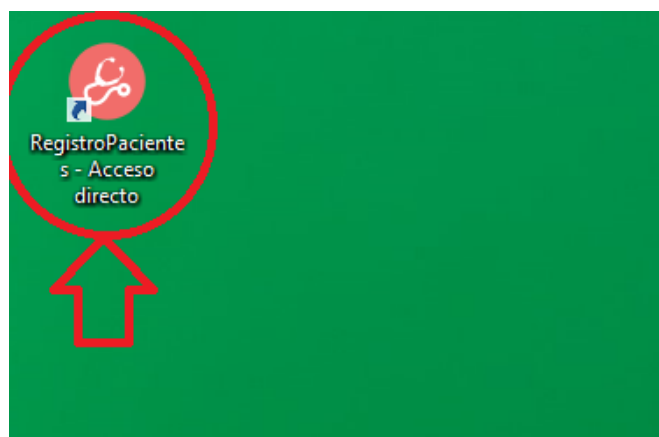
1) Cómo instalar el sistema

Una vez insertado el CD lo primero que usted debe hacer es dirigirse al ícono cuyo nombre es “Setup” y hacer doble click sobre él.



Se abrirá una pestaña en la cual debe elegir el idioma que utilizará durante la instalación. Luego debe elegir el directorio donde instalará el sistema (se recomienda dejarlo “por defecto”, es decir, dejando todo tal como está). Presione Siguiente y seguido a esto en Instalar. Por último, dé click en Finalizar y usted ya tendrá disponible el sistema para su utilización (cabe aclarar que todo el proceso de Instalación se encuentra guiado por el mismo instalador).

2) Cómo acceder al sistema



Primero clickeee en el ícono de acceso marcado.

Una vez accedido se abrirá una nueva ventana como se muestra a continuación en la cual debe ingresar su usuario y contraseña.

En el caso de:

- a) No tener un usuario y contraseña presione el botón “Regístrate”.
- b) Olvidar su contraseña, presione el botón ¿Olvidó su contraseña?



3) Cómo ingresar datos



Al clicar en “Ingreso de datos” se abrirá una nueva ventana como se muestra en la siguiente imagen:



a. Ingreso de pacientes (Paciente)



Al realizar click sobre la opción “Paciente” abrirá una ventana donde podrá ingresar los datos del paciente atendido. El código debe ser numérico. El apellido y nombres caracteres alfabéticos.

b. Ingreso de médico (Medico)



Al realizar click sobre la opción “Medico” se abrirá una ventana donde podrá ingresar los datos del médico. El código debe ser numérico. A continuación, deberá ingresar el nombre completo del médico (apellido y nombres) y por último la especialidad del médico (la cual se podrá seleccionar de la lista desplegable).

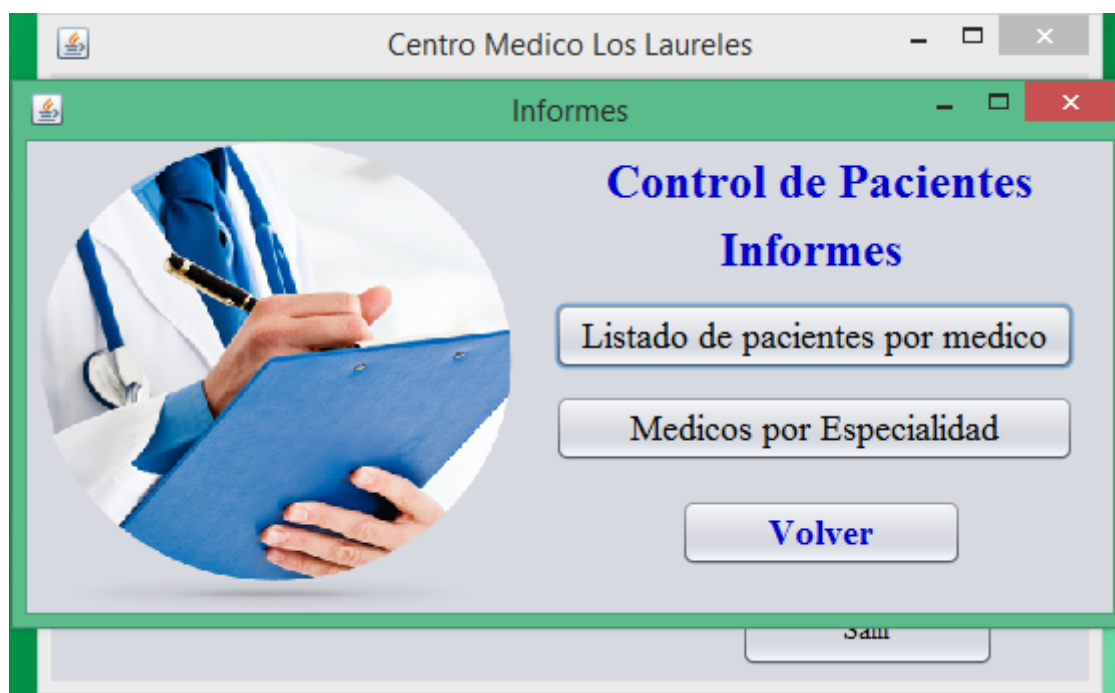
c. Ingreso de diagnósticos (Diagnostico)



Posterior a la carga del paciente y el médico deberá ingresar el diagnóstico que reflejará la situación actual del paciente atendido. Primero se deberá seleccionar el paciente dentro de la lista y luego el médico (ambos ya fueron cargados previamente). Por último, escriba la nota correspondiente al diagnóstico del paciente.

4) Visualizar informes

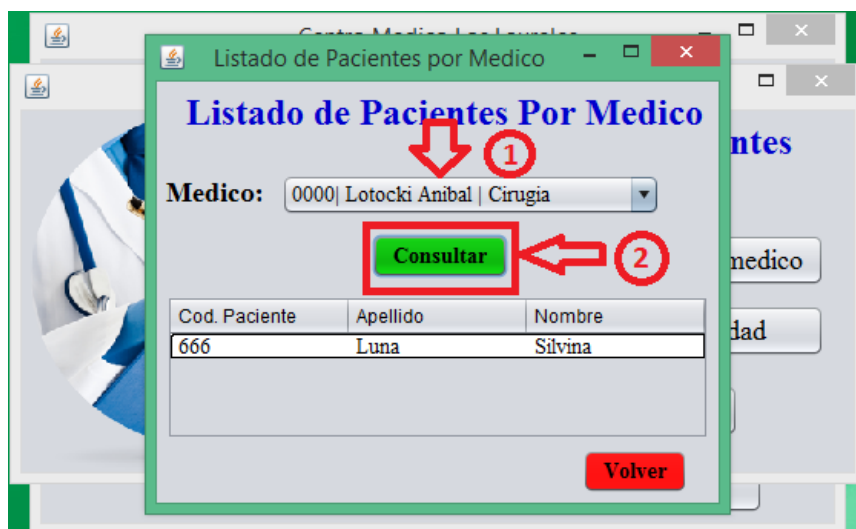
Si usted selecciona la opción de “Informes” en el menú principal se abrirá una ventana como la siguiente:



a. Listado de pacientes por médico

Si usted quiere visualizar los pacientes existentes por médico primero deberá seleccionar el médico dentro de la lista (siempre y cuando haya sido cargado previamente con sus correspondientes pacientes) y presionar en el botón “Consultar”

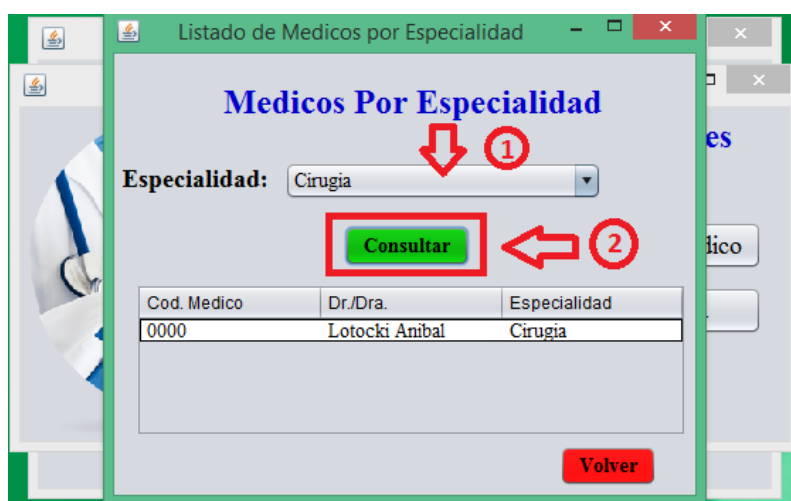
A continuación, se muestra una imagen con el procedimiento a modo de ejemplo:



b. Médicos por especialidad

Si usted quiere visualizar los médicos existentes pero filtrados por especialidad primero deberá seleccionar el tipo de especialidad dentro de la lista y por último presionar en el botón “Consultar”.

A continuación, se muestra una imagen con el procedimiento a modo de ejemplo:



5) Salir del sistema

Para salir del sistema puede ir al menú principal y presionar el botón “Salir” o simplemente presionar la “X” y cerrar la ventana.

Encuesta de Satisfacción de Usuario y Comparación

Realizamos la siguiente encuesta a 8 personas diferentes para saber su opinión sobre nuestro producto software:

- 1) Funcionabilidad
 - ¿Siente que el sistema es seguro con los datos que maneja? SI/NO
 - ¿El sistema le brindó siempre los resultados que esperaba? SI/NO
 - ¿El sistema cuenta con todas las funcionalidades que necesita? SI/NO
- 2) Usabilidad
 - ¿El programa lo ha ayudado a superar cualquier problema que ha tenido durante su uso? SI/NO
 - ¿Considera que el programa es atractivo visualmente? SI/NO
 - ¿Debió consultar la ayuda muchas veces durante la utilización del programa? SI/NO
- 3) Eficiencia
 - ¿El programa tarda mucho en otorgarle los resultados? SI/NO
 - ¿El programa funciona de igual manera al usarlo por tiempo prolongado a cuando se lo usa en periodos pequeños? SI/NO
 - ¿Su computadora funciona igual al utilizar el software que cuando no lo utiliza? SI/NO
- 4) Portabilidad
 - ¿Le costó mucho instalar el software? SI/NO
 - ¿Cuándo utiliza el programa, las demás aplicaciones funcionan de igual manera? SI/NO
 - ¿Pudo usar el software correctamente en el sistema operativo de su computadora? SI/NO
- 5) Fiabilidad
 - ¿El software en algún momento se detuvo inesperadamente? SI/NO
 - ¿Si este software se detiene, es fácil reiniciarlo? SI/NO
 - ¿El sistema tiene mensajes de error claros? SI/NO

Luego de obtener los resultados, realizamos los cálculos para obtener los puntajes por cada pregunta, y a su vez el puntaje por cada categoría de la encuesta.

A partir de los datos obtenidos, realizamos una comparación con nuestro modelo de calidad.

Los criterios utilizados para calcular los puntajes de cada pregunta fueron:

- 5 puntos por cada respuesta positiva (puede ser SI o NO dependiendo de la pregunta).
- 0 punto por cada respuesta negativa (puede ser SI o NO dependiendo de la pregunta).

Los resultados obtenidos de nuestro modelo de calidad son:

- Funcionabilidad: 4 puntos.
- Usabilidad: 4,33 puntos
- Eficiencia: 5 puntos.
- Portabilidad: 4 puntos.
- Fiabilidad: 4 puntos.

La encuesta realizada es la siguiente:

PREGUNTAS	E1	E2	E3	E4	E5	E6	E7	E8	CATEGORÍA
¿Siente que el sistema es seguro con los datos que maneja?	NO	SI	SI	SI	SI	SI	SI	SI	FUNCIONABILIDAD
¿El sistema le brindó siempre los resultados que esperaba?	SI	SI	SI	SI	SI	SI	SI	SI	
¿El sistema cuenta con todas las funcionalidades que necesita?	SI	SI	SI	SI	SI	SI	NO	NO	
¿El programa lo ha ayudado a superar cualquier problema que ha tenido durante su uso?	NO	SI	NO	SI	SI	SI	SI	SI	USABILIDAD
¿Considera que el programa es atractivo visualmente?	SI	SI	NO	SI	NO	NO	SI	SI	
¿Debió consultar la ayuda muchas veces durante la utilización del programa?	NO	NO	SI	SI	NO	NO	NO	SI	
¿El programa tarda mucho en otorgarle los resultados?	NO	NO	NO	NO	SI	NO	NO	NO	EFICIENCIA
¿El programa funciona de igual manera al usarlo por tiempo prolongado a cuando se lo usa en periodos pequeños?	SI	SI	SI	SI	NO	SI	NO	SI	
¿Su computadora funciona igual al utilizar el software que cuando no lo utiliza?	SI	SI	SI	NO	SI	SI	NO	SI	
¿Le costó mucho instalar el software?	NO	NO	SI	SI	NO	NO	NO	NO	PORTABILIDAD
¿Cuándo utiliza el programa, las demás aplicaciones funcionan de igual manera?	SI	SI	SI	SI	SI	SI	NO	SI	
¿Pudo usar el software correctamente en el sistema operativo de su computadora?	SI	SI	SI	SI	SI	SI	SI	SI	
¿El software en algún momento se detuvo inesperadamente?	NO	NO	NO	NO	SI	NO	SI	NO	FIABILIDAD
¿Si este software se detiene, es fácil reiniciarlo?	SI	SI	SI	SI	SI	SI	SI	SI	
¿El sistema tiene mensajes de error claros?	NO	SI	SI	SI	SI	SI	SI	SI	

Puntajes promedios obtenidos por pregunta:

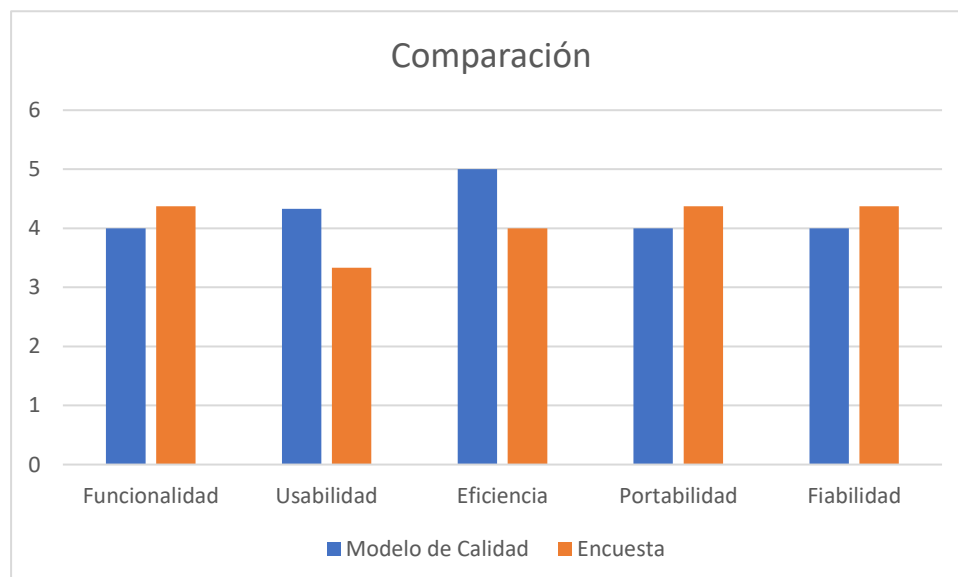
- Pregunta 1: 4,375 puntos
- Pregunta 2: 5 puntos
- Pregunta 3: 3,75 puntos
- Pregunta 4: 3,75 puntos
- Pregunta 5: 3,125 puntos
- Pregunta 6: 3,125 puntos
- Pregunta 7: 4,375 puntos
- Pregunta 8: 3,75 puntos
- Pregunta 9: 3,75 puntos
- Pregunta 10: 3,75 puntos
- Pregunta 11: 4,375 puntos
- Pregunta 12: 5 puntos
- Pregunta 13: 3,75 puntos
- Pregunta 14: 5 puntos
- Pregunta 15: 4,375 puntos

Puntajes obtenidos por categoría:

- Funcionabilidad: 4,375 puntos.
- Usabilidad: 3,33 puntos
- Eficiencia: 4 puntos.
- Portabilidad: 4,375 puntos.
- Fiabilidad: 4,375 puntos.

Comparación:

	Modelo de Calidad	Encuesta
Funcionalidad	4	4,375
Usabilidad	4,33	3,33
Eficiencia	5	4
Portabilidad	4	4,375
Fiabilidad	4	4,375

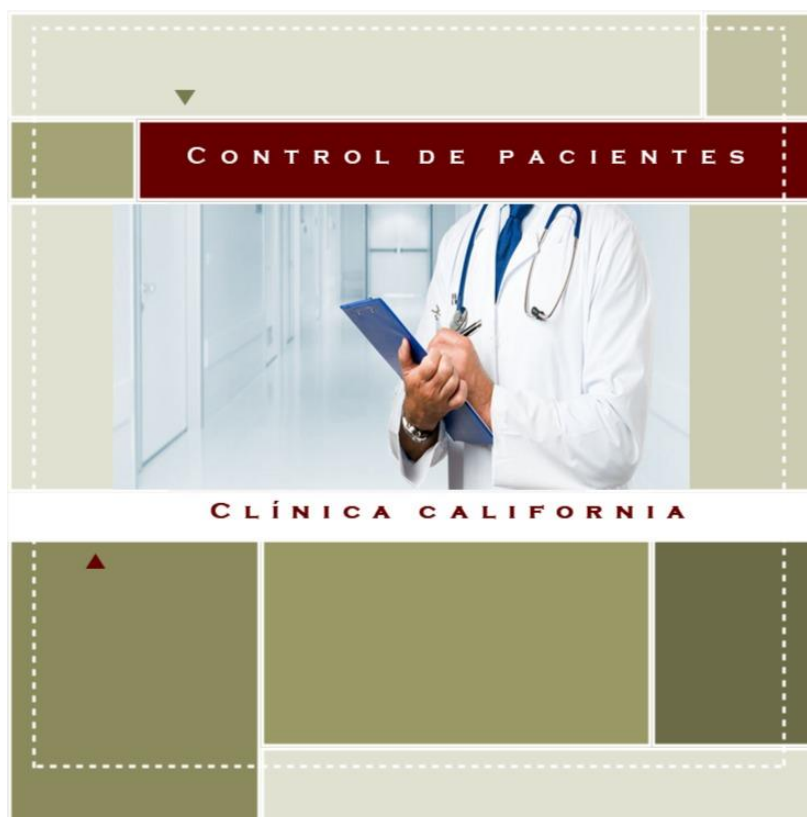


Hemos obtenidos resultados muy similares entre el modelo y la encuesta, tanto en Funcionalidad, Portabilidad y Fiabilidad.

En los dos restantes, Usabilidad y Eficiencia, hemos tenido una pequeña diferencia a favor del modelo, ya que son categorías en las que en el modelo de calidad pudimos cumplir prácticamente en todos los puntos, y en la encuesta quizás realizamos alguna pregunta que obtuvo opinión dividida de los encuestados. También influyó que el modelo de calidad contiene puntos mucho más detallados, y la encuesta son preguntas que quizás no tengan el grado de detalle adecuado, pero que es necesario preguntarlas de ese modo teniendo en cuenta el público al que es dirigido la misma.

Cubierta de CD (ISO 9127)

- Anverso de la Tapa:



- Reverso de la Tapa:



- Ploteo de CD:

