

Computer Program for Managing Medical Appointments

Diseño de un Programa de Gestión de Citas Médicas

Pedro Huacho, Maria Loza, Jose Barza, Alvaro Alfaro Orlando

Course: CC201 Object Oriented Programming

Engineering Physics Program

College of Sciences

National University of Engineering

ABSTRACT

In this project a computer program is designed and implemented for managing medical appointments in a health center in a simple and friendly way.

The program structure, the variables, the data base, and the classes have been properly defined to attain the proposed objectives.

The program allows the creation, modification, validation and deleting of appointments using information of the medical specialty, the patient, date and hour of the appointment.

The program provides the doctor an Excel report with the appointments continuously updated.

Use case diagrams and class diagrams have been elaborated and integrated with a SQL data base to design and implement an information system supporting physician and health center activities.

INTRODUCCIÓN

El uso de un servicio médico es muy común en todas partes del mundo. Millones de personas alrededor del mundo se atienden diario en alguna clínica. Ya sea por alguna emergencia, prevención o tratamiento.

La primera parte del sistema médico es la reserva de una cita con el especialista. Esta debe ser oportuna y sin mayores complicaciones. Es por eso que con lo globalizado de nuestros tiempos, se ve necesario la implementación de algún software que ayude en este aspecto.

Actualmente existe algún software orientados a los doctores para ayudarles a la organización de sus citas, tales son por ejemplo: Gestor Citas de Médico 1.3 y el Consultas Médicas 1.1.23. Dichos software presentan una plataforma en la cual el médico ingresa los datos de las citas próximas que tiene.

Sin embargo dichos programas como algunos otros, no le brindan al médico las citas que tiene de manera automática. Esta es la razón por la cual se presenta este proyecto, con el objetivo de mediante el uso de la base de datos de la clínica, se pueda presentar al médico la información que necesaria saber sobre las citas venideras y las que ya ocurrieron.

De la misma manera el programa que se desarrolló permite que el manejo de las citas desde el momento de su creación sea sencillo.

Para la realización de este proyecto se usaron diversas herramientas tecnológicas; predominando entre ellas el uso del lenguaje de programación orientada a objetos: java.

PRESENTACIÓN DEL PROBLEMA

Desde hace algunos años el mundo de la medicina se ha sumado con la tecnología de muchas formas, y en todos los casos esa unión ha sido favorable para la medicina misma.

Esta unión se ha visto en el aspecto técnico de la misma medicina; pero vemos que muchos procesos de sistema médico aún no se han automatizado.

Actualmente en muchos centros médicos se complica el proceso en el cual el paciente separa una cita con el médico, ya sea por la demora o complejidad del proceso, o bien por la gran afluencia de personas con el mismo propósito.

Si esta situación continúa así, las personas dejarán de ir a los centros médicos; ya que encontrarían más malestar que alivio al momento de separar su cita, personalmente o por teléfono. Ocasionándose más problemas de salud en la población, cuya atención médica sería a destiempo de cuando se pudo haber tratado con facilidad.

Los médicos, por otra parte, tienen problemas para poder obtener la información de todas las citas programadas que tengan, lo que ocasionaría pérdida de información al médico, que en muchos casos es perjudicial para el mismo.

OBJETIVOS

Objetivo General: Desarrollar un programa de escritorio que se encargue de la gestión de consultas médicas.

Objetivos Específicos:

Determinar una manera adecuada en la cual el paciente pueda separar su cita.

Determinar de qué manera se le puede presentar al médico la información de sus citas.

Aplicar los conceptos de lenguaje orientado a objetos en la elaboración del programa.

DESCRIPCIÓN DE LA SOLUCIÓN

BASE DE DATOS:

El programa que se desarrollará estará orientado a una clínica que cuente con una base de datos de los horarios de los doctores, los doctores y la especialidad que se atienden en dicho lugar. Es por eso que se determinó las tablas necesarias para el desarrollo del programa.

En la figura 1.1, 1.2, 1.3 y 1.4, se muestra las tablas con las que se a trabajo. Su desarrollo se realizó en la plataforma de base de datos SQL de Microsoft.

INTERFACES:

Se diseñó las interfaces con la intención de que las mismas sean atractivas para el usuario, y muestren claramente el funcionamiento del programa.

Se determinó con las mismas las 3 funciones principales que el programa se encargará de hacer. Figura 2. A continuación se detalla cada una de las funciones con su respectiva interfaz.

CITAS:

Esta función se encarga del manejo de las citas en su forma más básica.

Dentro de esta función se encontrarán 3 funciones más .Figura 3. Las cuales detallamos a continuación.

Creación de Cita. Se determinó que la forma más sencilla de crear citas era empezando por la especialidad y el médico; ya que si se empieza con los datos del paciente y este no está de acuerdo con su cita al final, sería una pérdida de tiempo para el usuario que se dedique a hacer la creación.

Por los mismo, se creó una secuencia de interfaces de tal manera que primero se ingrese la especialidad a la cual se quiere consultar, luego el médico, y finalmente se muestre los horarios disponibles. Luego de elegir el horario que más le convenga al paciente, se piden solo su nombre y su teléfono. Todos estos son los datos relevantes para el médico, sobre las citas.

Modificación de Cita. Se analizó la posibilidad en la cual una cita ya creada necesite ser modificada, es por eso se creó esta función.

En la interfaz de esta función nos permite cambiar de Teléfono, en el caso que el paciente haya cambiado su teléfono o haya habido algún error al momento de la creación de la cita.

También se puede cambiar el médico y el horario de la cita. Se investigó acerca del cambio de especialidad, encontrándose que este no es permitido en una clínica. Si ocurriese el caso, dicho cambio se puede hacer a través de la eliminación y creación de una nueva cita.

Eliminar: esta función permite eliminar las citas que se vea necesarias a través del código de la cita. Luego de ingresar el código de la cita, se mostrará la información de la cita para asegurarse que esa sea la cita correcta que se desea eliminar.

Si por alguna razón no existe acceso al código de la cita que se desea modificar o eliminar, se creó una interfaz de búsqueda. Figura 4. En ella solo es necesario ingresar el nombre del paciente, y se mostrará luego todas las coincidencias. Esta interfaz se enlaza tanto en las funciones de modificar y eliminar como en la segunda función principal que se pasa a detallar a continuación.

VALIDAR PAGO:

Esta función sirve para cambiar el estado de la cita de “no pagado” a “pagado”, ya que en el momento de la creación de la cita, esta tiene automáticamente el estado de “no pagado. Esto se debe a que el paciente puede realizar el depósito o pago por su cita, en el momento que vea conveniente antes de la realización de su cita. Luego lleva su recibo para que se cambie el estado de su cita a “pagado”.

Se determinó que esta función debe ser una función principal y estar separada de las demás funciones que gestionan las citas. Esto se debe a que se piensa para esta función un manejo especial con la base de datos.

Cada clínica maneja su propia base de datos con la información clínica del paciente, y también con información de los ingresos que recibe. Es por eso que esta función se creó pensada a una posible

conexión con el sistema de pago que pueda existir en la clínica.

A esta función también se la ha enlazado la interfaz de búsqueda de la Figura 4

CONSULTAR CITA:

Esta función es exclusiva de uso para el médico. El médico necesitará poner su código y luego el programa le permitirá visualizar todas las citas asignadas al médico. Esto permitirá al médico obtener la información básica de una cita.

También puede exportar dicha información a una hoja de cálculo en Excel.

CONCEPTOS DE LENGUAJE ORIENTADO A OBJETOS

Se determinó que se usaría varios conceptos del lenguaje orientado a objetos, si no todos; pero la mayoría y los más importantes.

Se usó los siguientes conceptos.

Clases: Esta es una definición fundamental en el lenguaje orientado a objetos, se vio conveniente elaborar las clases de paciente, médico, cita y especialidad. Ya que con estos objetos son con los que se trabajaron a lo largo del desarrollo del programa.

Se definió atributos necesarios en la creación de las clases, usándose también modificadores de acceso para seguridad informática.

Se elaboró más clases que contenían los diferentes métodos los cuales se usarán en el programa.

Paquetes: Para la mejor organización de todas las clases que se crearon, teniendo en cuenta que no todas las clases se comportan de la misma manera. Se organizó las clases en diferentes paquetes.

Para esta organización se usó los conceptos de casos de uso, con las que se determinó las principales funciones que el programa debería realizar atendiendo los requerimientos del usuario. Se establecieron los paquetes y sus relaciones.

Las clases que se usaron para instanciar los objetos con los que se iba a trabajar principalmente se pusieron en un paquete de modelo-bean.

Se usó el concepto de interfaces. Con la creación del modelo-dao. En donde se indicaban todos los métodos que serían usados a lo largo del desarrollo del programa; pero solo se indicaban cuáles serían y que las mismas tratarían del manejo con la base de datos. En el paquete modelo-dao-jdbc se creó otra clase en la que se describían la función de dichos métodos.

Se creó otro paquete en el cual se declaran clases que tiene métodos que se encargan de generar tablas, que luego serán mostradas; por ello a dicho paquete se le llamó el paquete de vistas.

Finalmente, el paquete control sería en el cual se indicaría todo el proceso que el programa realizará; usando los métodos ya definidos en otro paquete, Este paquete contendría principalmente clases que usan la librería swing; para la creación de la interfaz gráfica.

Encapsulamiento: Por tema de seguridad informática se realizó el encapsulamiento en las clases del paquete modelo-dao, que son las que al momento de instanciarse nos permitirá el manejo de la información de la base de datos.

Librería Swing: El programa se trabajó en la plataforma NetBeans.

En la misma se desarrolló la interfaz gráfica del programa, usando puramente código. Lo que permitió un mejor desarrollo del programa en su parte operacional.

Excepciones: El uso de este concepto fue muy importante en el desarrollo del programa. Con el uso de las excepciones se pudo controlar muchos errores que se generaron en las pruebas del programa.

Con ella se pudieron crear muchas ventanas de error, pero permiten al programa continuar con su ejecución.

Uso de base de datos: Se diseñó el programa de tal manera que pueda ser usado con una base de datos real de una clínica. Para ellos las pruebas se

realizaron con una base de datos en la plataforma para base de datos SQL de Microsoft.

Se realizó el manejo de los datos de la base de datos a través de las funciones que ofrece la librería gratuita del programa, que se descargó su la página web oficial.

Uso de la librería POI: Se decidió incluir una función en la cual el programa pudiera exportar los datos de las citas en una hoja de cálculo de Excel; para que el médico pueda usarlo a su conveniencia.

Para que esto fuera posible se usó las funciones que ofrecen la librería POI, del proyecto Apache. Esta librería ofrece diversas funciones para la importación y exportación de documentos de Excel, como de otros programas de Microsoft.

RESULTADOS

En la realización y pruebas del programa se obtuvieron los siguientes resultados:

Se Desarrolló el programa plenamente con todas las interfaces que se diseñaron previamente. Cada una enlazada de manera correcta entre sí.

La secuencia final del programa es la descrita a continuación.

Se ingresa al programa, se muestra el logo del programa mientras se carga el programa.

Luego el usuario encuentra 3 opciones: Citas, Validar Pago y Consultar Citas.

Al seleccionar Citas, se le mostrará 3 opciones más Crear, Modificar y Eliminar.

Al seleccionar Crear el usuario deberá ingresar primero la especialidad en la cual quiere registrar la cita. Esta dependerá de las especialidades que tenga la clínica. Luego continuará.

Lo siguiente que se pedirá es el médico con el que se desea atender; para ellos el programa le mostrará una lista con todos los médicos de la especialidad que seleccionó previamente. Al escoger al doctor puede continuar.

Se mostrará a continuación un cuadro en el cual puede observar todos los horarios del médico según el día y la hora, los cuales estarán disponibles o no.

Solo podrá elegir uno que contenga un estado disponible. Si elige uno que no está disponible, simplemente no podrá continuar.

Si no existe un horario que el paciente quiera seleccionar, puede volver atrás para seleccionar otro médico o simplemente cancelar la operación.

De continuar se le mostrará a continuación dos cuadros en los que deberá ingresar el nombre y el teléfono del paciente.

Finalmente se le mostrará un cuadro de confirmación con todos los datos que ingresó previamente. Adicionalmente se le mostrará también el costo de la cita. Este costo saldrá de la base de datos y dependerá de la especialidad en la que se desea atender.

De tener un seguro, o requerir un descuento esto deberá ser consultado por el paciente en la caja de la clínica o cuando hace el depósito.

Si se requiere alguna modificación de la cita se usará la opción de Modificar, en ella se mostrará todos los datos de la cita, y el paciente puede cambiar tu teléfono, el horario, o el doctor con el que se quiere atender.

Con la opción Eliminar se podrá eliminar la cita. Antes de la eliminación se mostrará un cuadro con los datos de la cita a la cual se a seleccionado.

Para estas dos opciones se requiere el código de la cita; pero si no se tiene muy claro de cuál es el código de la cita que el usuario tiene en mente modificar o eliminar puede seleccionar la opción de búsqueda.

En ella solo tiene que ingresar el nombre completo o alguna referencia del mismo. Y el programa le ofrecerá un cuadro con todas las coincidencias. Se le mostrará el nombre del paciente, especialidad, doctor, fecha, hora y código de la cita.

Una vez que se tenga claro el número de la cita el usuario podrá regresar para continuar con operación.

Una vez que se haya hecho el pago, puede ser días después, se podrá validar el pago. Para el cual se selecciona la opción de Validar Pago, del menú principal.

Nuevamente se proporciona la opción de búsqueda por si no se tiene en claro el código de la cita a la cual se quiere validar.

Se mostrará un cuadro que indicará si la cita ingresada está o no pagada. De estar pagada no se podrá validar otra vez. Pero de no estar pagada puede validarse con el botón de validar.

La última opción es de uso para los médicos. El médico debe conocer su código, que estará almacenado también en la base de datos. Al momento de ingresar su código podrá visualizar todas las citas que tiene. También podrá exportar las mismas a una hoja de cálculo de Excel, que se abrirá automáticamente al seleccionar el botón de exportar datos a Excel.

En esta hoja se mostrará todas las citas que antes pudo visualizar a través del programa. Esta hoja de cálculo, tendrá resaltado los campos de la cita, como son: código de cita, nombre de paciente, teléfono, fecha, hora y estado de pago.

Las citas también serán enumeradas desde el número uno, para que se pueda saber la cantidad de citas que se están mostrando.

En las pruebas que se realizaron se encontraron una serie de errores que fueron inmediatamente corregidos para que el producto final funcione como se había diseñado.

El diseño fue enriquecido en todo momento, al surgir nuevas ideas mediante se desarrollaba el programa.

CONCLUSIONES

Las conclusiones que podemos rescatar del presente proyecto son:

La organización de las clases que se desarrollaron en paquetes, permitió un mejor desarrollo general del programa ya que permitió ver al programa de manera segmentada y ordenada.

El uso de las excepciones fue la mejor manera de solucionar problemas al momento de correr el programa, ya que muchas veces estas permitían controlar un problema más fácilmente que tratar de resolverlo para que funcionara bien.

El programa de escritorio que se desarrolló no es complicado y la información que requiere no es extensa, solo la necesaria para realizar una buena gestión de la misma.

BIBLIOGRAFÍA

[1] Yamil Armando Cerquera. Fundamentos de Programación.

[2] Robert J. Liguori. Edward G. Finegan.

Sun Certified Java Associate Study Guide.

[3] Deitel y Deitel. Como programa Java Novena Edición.

Tablas e Interfaces

Figura 1.1 Tabla de Citas

Cod_cita	Cod_especiali...	Nombre_pacie...	Telefono	Fecha	Hora	Cod_medico	Estado_pago
1	MED170	Edda	1111111	2014-06-01	11:00:00	101	No pagado
2	MED170	Madeleyne	6666666	2014-06-01	13:00:00	101	Pagado
6	MED170	Raul	954545453	2014-06-02	10:00:00	101	No pagado
7	CAR150	Alejandra	111222333	2014-06-01	13:00:00	201	Pagado

Figura 1.3 Tabla de Especialidades

Cod_especiali...	Especialidad	Precio
CAR150	Cardiologia	150.0000
DER150	Dermatologia	150.0000
EST190	Estomatologia	190.0000
MED170	Medicina General	170.0000
PED180	Pediatria	180.0000

Figura 1.2 Tabla de Médicos

Cod_medico	Nombre_medi...	Especialidad
101	Daniel Muñoz	Medicina General
102	Diana Salas	Medicina General
201	Leonardo Cabr...	Cardiologia
202	Carolina Chang	Pediatria
203	Alfonso Balaguer	Dermatologia

Figura 1.2 Tabla de Horarios

Cod_horario	Cod_medico	Fecha	Hora	Estado
10114060109	101	2014-06-01	09:00:00	OCUPADO
10114060110	101	2014-06-01	10:00:00	OCUPADO
10114060111	101	2014-06-01	11:00:00	DESOCUPADO
10114060112	101	2014-06-01	12:00:00	OCUPADO
10114060113	101	2014-06-01	13:00:00	OCUPADO
10114060210	101	2014-06-02	10:00:00	OCUPADO
10114060211	101	2014-06-01	11:00:00	DESOCUPADO
10114060213	101	2014-06-02	13:00:00	OCUPADO

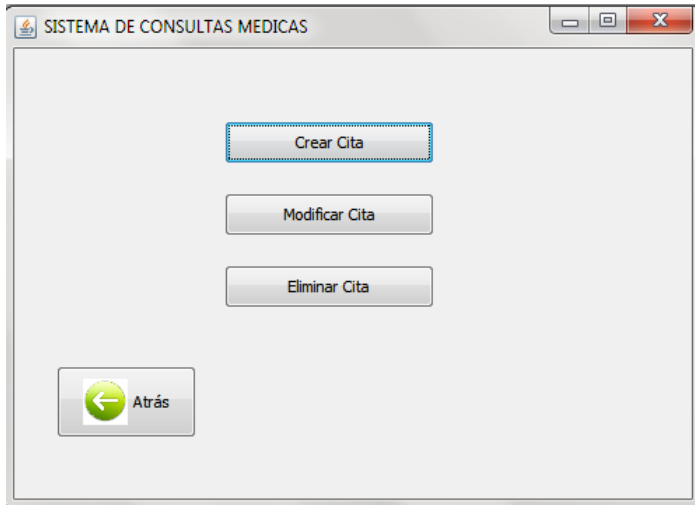


Figura 2. Interfaz principal



Figura 4 Interfaz de Búsqueda

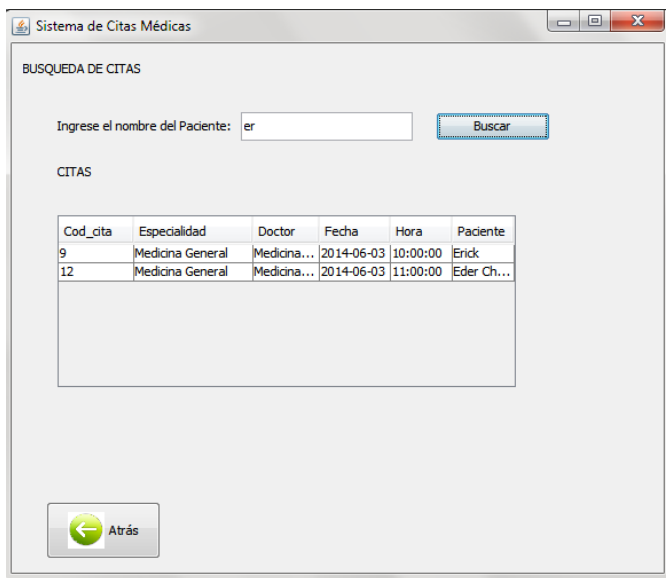


Figura 3 Interfaz de Citas