

Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Programación Orientada a Objetos

Proyecto Programado 1

Prof. Mauricio Avilés Cisneros

Andrea Abarca Baltodano (2015088898)

Darío Monestel Corella (2014073400)

Jafeth Rivas Hernández (2015092155)

I periodo

2016

Barbería

1. Introducción	1
1.1 ¿Por qué se hace el proyecto y qué se incluye?	2
2. Presentación y análisis del problema	2
2.1 Entendimiento del problema	3
2.2 Cómo se va a resolver el problema	5
2.3 Implementación y análisis de los resultados	6
3. Diagrama de clases	7
4. Conclusiones	8
5. Recomendaciones	9
6. Referencias	11

1. Introducción

1.1 ¿Por qué se hace este proyecto y qué se incluye?

Desde los inicios de las ciencias de la computación, la manipulación de objetos, se ha dado la necesidad de ordenar objetos con el fin de que puedan ser manipulados y comprendidos de manera más fácil por los seres humanos en términos computacionales, esto se ha desarrollado de una buena forma en las últimas décadas con la aparición de diversos lenguajes de programación orientados a este paradigma así como ambientes de desarrollo para representar de una manera gráfica lo implementado por los desarrolladores de software, esto ha ayudado a que los usuarios que no tengan muchos conocimientos informáticos interactúen de una manera más sencilla con estos programas.

El objetivo del presente proyecto será el desarrollo de un sistema de citas para una barbería, este sistema será una aplicación de escritorio para la programación de citas, que permita al encargado del local crear citas, ver la programación de las citas y enviar notificaciones vía correo electrónico a sus respectivos clientes. Aunque el proyecto estará orientado a los servicios brindados por una barbería, la aplicación podría ser utilizada para otros casos similares. Cabe agregar que al ser esta aplicación de escritorio su manejo será limitado a los miembros de la organización y requieren que el cliente se comunique con la organización correspondiente.

Lo anterior nos lleva a requerir diversas herramientas para el desarrollo de software pero hablando de una forma más específica se hará uso de diagramas UML el cual ayudará en el análisis y diseño del problema a desarrollar. Describiendo lo anterior se va a requerir un diagrama de casos de uso, un diagrama de clases y como buena práctica un prototipo para la interfaz de usuario.

Para desarrollar este sistema se hará uso de Java implementando el paradigma Orientado a objetos y para su interfaz se usará Java Swing.

2. Presentación y análisis del problema

2.1 Entendimiento del problema

Para la elaboración de este trabajo es necesario conocer la teoría de Programación orientada a objetos, es decir se requiere de un dominio sobre modelado UML para desarrollar un modelo de clases detallado con sus respectivas relaciones en cual ayudará a la comprensión del proyecto a realizar, así como conocimientos para desarrollo de software.

Qué se debe resolver

Como primer paso se deberá diseñar un diagrama de clases, en cual tomará algunos datos del diagrama de casos de uso el cual capta los requerimientos del sistema y las funcionalidades del programa.

En el segundo paso se deben implementar las clases del sistema contempladas anteriormente en código.

Luego se debe desarrollar la interfaz de usuario en la que se debe agregar los diferentes componentes funcionales y de usabilidad para el manejo de del programa.

Profundizando en este último paso se debe crear una ventana principal que mostrará las diferentes funcionalidades de la aplicación entre ellas tenemos: cita, cliente, tipo de servicio, sección de ayuda y una lista de espera. Contemplando cada punto funcional tenemos que el área de cliente que requiere una sección de creación y modificación de clientes que será usada por el usuario del sistema.

Luego se debe implementar la sección de cita la cual debe tener una fuerte relación con la mencionada anteriormente para la respectiva creación de citas y su respectiva área para editar los datos, cabe agregar que en esta funcionalidad se debe incorporar el envío de notificaciones a los respectivos clientes y una sección de confirmación por parte del cliente para ser atendido a la hora deseada por el mismo.

En el módulo de tipo de servicio, se mostrará una descripción de los servicios ofrecidos en este caso de la barbería, seguidamente se implementara una lista de espera que mostrará un desglose de la lista de clientes que van a requerir un servicio específico, sus datos y la hora que serán atendidos. Además se debe implementar una sección para visualizar el horario de atención por parte del administrador del local, se debe destacar que es indispensable realizar una investigación para la implementación en el caso de finalizar la aplicación los datos deben guardarse en disco, de esta forma al volver a ejecutar la aplicación debe aparecer la información tal como estaba antes de salir la última vez que fue utilizada.

Otro de los problemas a contemplar es la validación de los números de teléfono y los e-mails de los clientes para ser incorporados a la aplicación. Otra situación planteada es la visualización de un calendario para que el usuario observe de forma rápida la elección de la fecha para notificar la cita.

2.2 Cómo se va a resolver el problema

Como se explicó anteriormente se implementó un diagrama de clases para modelar el sistema a desarrollar, este incluye atributos, métodos y las relaciones entre las clases para facilitar la construcción del sistema para la interacción de las clases.

La inserción de clientes, citas, servicios y demás, así como las operaciones de cada uno de ellos se maneja por medio de listas con arreglos, que se crean en el constructor, donde además se escriben los métodos que manejan datos, desde la interfaz se llama cada método para hacerlo funcional en la pantalla.

En el guardado en disco se utiliza la clase File con las librerías:

- .io.FileInputStream
- .io.FileOutputStream
- .io.ObjectInputStream
- .io.ObjectOutputStream
- .io.Serializable

Para la visualización del calendario se usó una librería externa llamada jcalendar que luego de ser incorporada al ambiente de desarrollo se tuvo que realizar unas configuraciones en el Swing para importar las funcionalidades de la librería.

Para el envío de correos desde la aplicación se va hacer uso de una librería llamada java mail que es un paquete externo, la cual va a permitir que se envíen e-mails desde la aplicación, para ello se requiere de una cuenta preferiblemente de gmail, en la que realizaremos unas configuraciones como la verificación en dos pasos y el establecimiento de una contraseña de aplicación indispensable para su correcta función.

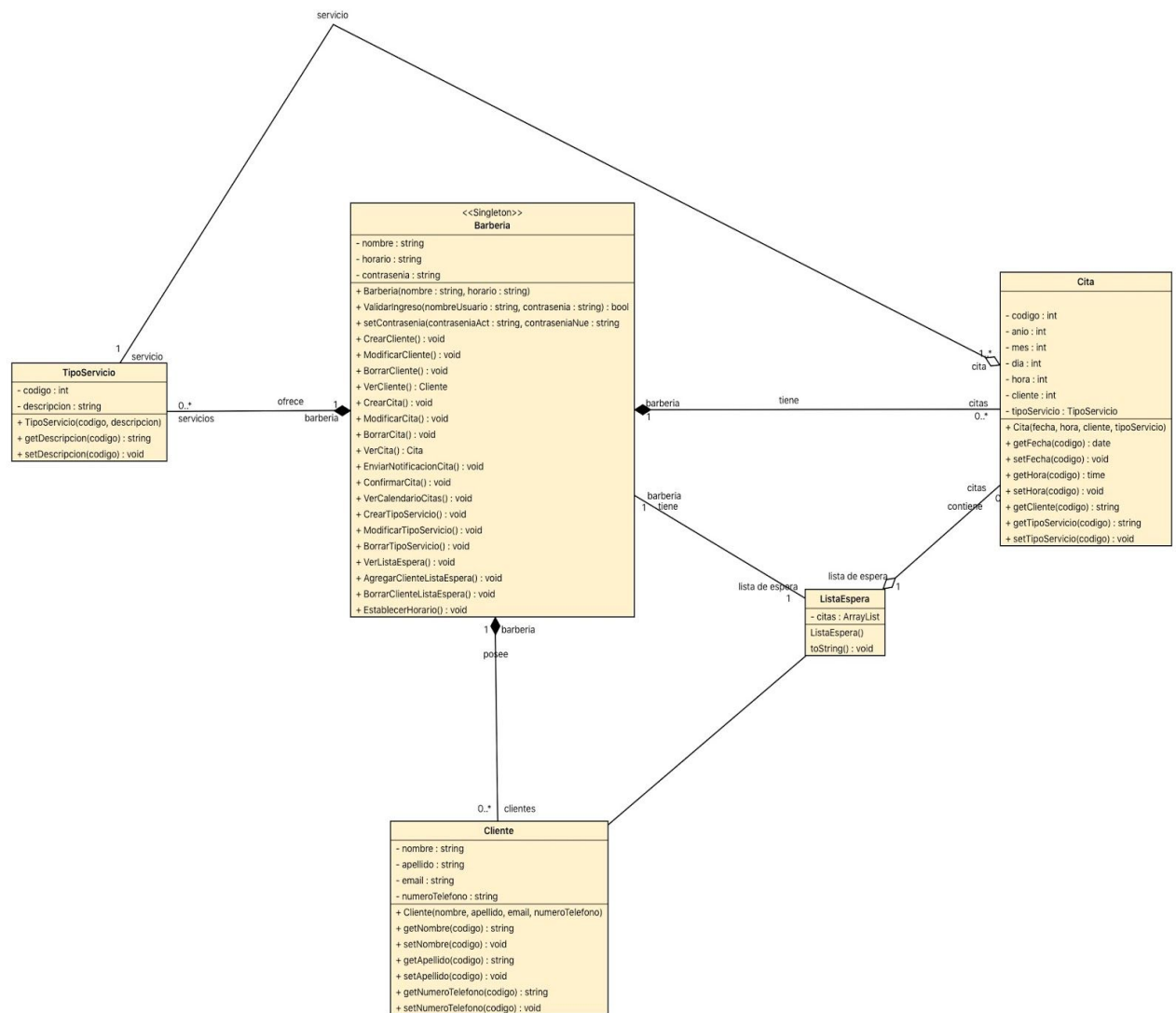
2.3 Implementación y análisis de los resultados

La codificación del proyecto se realizó con el lenguaje de programación Java en su última versión estable.

Se logró implementar el envío de notificaciones mediante la librería de Java mail a los clientes que necesiten una lista, además se implementó el calendario en el área de citas para ser visualizado por el usuario, además se realizó el guardado en disco para mantener los archivos almacenados por el usuario durante la ejecución.

Falta la implementación de la lista de espera e incorporar la funcionalidad de guardar en disco entre varias funcionalidades entre las pestañas.

3. Diagrama de clases



4. Conclusiones

- La documentación interna es de suma importancia, con el fin de hacer el código más entendible y aumentar la extensibilidad del programa.
- El entendimiento de la teoría básica de modelado de software mejoró el proceso de realizar este proyecto.
- La biblioteca gráfica Swing es sencilla de utilizar, sobre todo para la creación de múltiples ventanas en una aplicación.
- La investigación sobre las funcionalidades de algunas librerías externas ayudó a resolver exitosamente algunos problemas requeridos por este proyecto.
- El rendimiento de los algoritmos está en función de la computadora en la que se ejecute y depende de sus características de hardware (procesador, memoria ram, almacenamiento) y software (sistema operativo).
- Existe la necesidad de crear algoritmos híbridos para suplir las necesidades del mundo real, ya que la naturaleza de los algoritmos en la programación orientada a objetos tenemos implementaciones originales que no siempre responden en tiempos de ejecución.

5 Recomendaciones

- Definir claramente el problema o necesidad y los objetivos del proyecto. No existe un mayor logro en el área de proyectos que ver que la misma marcha por buen camino, cumpliendo los objetivos que se tenían propuestos.
- Estar motivado (as). Tener presente que el estar motivado es clave para el desarrollo de un proyecto y constituye uno de los factores principales de todo programador. No debe olvidarse que este punto lleva al éxito o al fracaso de un proyecto.
- Llevar a cabo un buen levantamiento de información especialmente a lo que respecta a la situación en la que se encuentre el equipo del proyecto; ya que existen casos en donde se desarrollan proyectos de software donde los conocimientos del individuo son inferiores a los requeridos, este problema se debe a la ausencia de alguna investigación previa.
- Identificar a tiempo posibles fuentes de problemas, este punto va a permitir reaccionar ante cualquier eventualidad y no perder el ritmo de la programación.
- Plasmar el proyecto de la forma más entendible posible de tal manera que quien lea información capte de forma correcta lo que realmente se quiere hacer.
- Utilizar siempre que sea posible el paradigma orientado a objetos para facilitar la solución a los problemas por resolver.

- Hacer una documentación interna explicativa y entendible.
- Utilizar algún software para el manejo del control de versiones (Git, Mercury, entre otros).
- Hacer uso de las normas de clean code para fomentar buenos hábitos de programación para ser aplicados en futuros proyectos.

6. Referencias

Wikipedia, la enciclopedia libre. Swing - Wikipedia, la enciclopedia libre.
Retrieved from [https://es.wikipedia.org/wiki/Swing_\(biblioteca_gr%C3%A1fica\)](https://es.wikipedia.org/wiki/Swing_(biblioteca_gr%C3%A1fica))

Overview,(JavaMail). Documentation JavaMail API
<https://javamail.java.net/nonav/docs/api/>

GreekTheory, Swing (interfaces gráficas)- Mario Pérez Esteso.
<https://geekytheory.com/tutorial-14-java-swing-interfaces-graficas/>

Read and write objects, Java fundament
<http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/archivos/objetos.htm>

Swing, la solución actual de Java para crear GUIs
<http://users.dcc.uchile.cl/~lmateu/CC60H/Trabajos/edavis/swing.html>

Toedter, JCalendar 2016 Kai Toldter <http://toedter.com/jcalendar/>