****

**UNIVERSIDAD TECNOLÓGICA DEL PERÚ**

**Carrera Profesional De Ingeniería De Sistemas E Informática**

**SISTEMA PARA LA GESTIÓN DE RESERVA DE CITAS**

**MÉDICAS PARA LA CLÍNICA “LAS AMERICAS”**

**AVANCE DEL TRABAJO FINAL**

**ALUMNOS:**

HUANCAHUARI CHANG, ARIAM EDER U20208963

TAMAYO ALFARO, FERNANDO ANDRÉ U20209263

LLERENA MENDOZA, FERNANDO JOSÉ U20205043

ROJO MORÓN, PIERO ALESSANDRO U20217841

FALCONI RODRÍGUEZ, RENZO DANIEL U20217199

CONDORI CUBA, VÍCTOR CRUZ U20217570

**Ica – Perú**

**2022**

INDICE

[1. ASPECTOS GENERALES 4](#_Toc104739246)

[1.1. Definicion del problema: 4](#_Toc104739247)

[1.1.1. Descripción del problema: 4](#_Toc104739248)

[1.1.2. Formulación del problema 4](#_Toc104739249)

[1.2. Definición de objetivos 4](#_Toc104739250)

[1.2.1. Objetivos generales 4](#_Toc104739251)

[1.2.2. Objetivos específicos 5](#_Toc104739252)

[2. MARCO TEORICO 5](#_Toc104739253)

[2.1. Fundamento teórico 5](#_Toc104739254)

[2.2. Planificación del trabajo 5](#_Toc104739255)

[2.3. Producto obtenido 6](#_Toc104739256)

[3. ANÁLISIS DE REQUISITOS 6](#_Toc104739257)

[3.1. Visión General 6](#_Toc104739258)

[3.2. Requisitos funcionales 7](#_Toc104739259)

[3.3. Requisitos de la interfaz de usuario 7](#_Toc104739260)

[3.3.1. Identificación de las restricciones técnicas 7](#_Toc104739261)

[3.3.2. Perfiles de usuario 7](#_Toc104739262)

[3.3.3. Documentación de las tareas 8](#_Toc104739263)

[4. TECNOLOGIAS EMPLEADAS: 9](#_Toc104739264)

[4.1. NetBeans 9](#_Toc104739265)

[4.2. Java 9](#_Toc104739266)

[4.3. MySQL 10](#_Toc104739267)

[4.3.1. Características de MySQL 10](#_Toc104739268)

[4.4. JFrame 11](#_Toc104739269)

[4.4.1. Objeto de control JTextField 11](#_Toc104739270)

[4.4.2. Métodos más usados: 11](#_Toc104739271)

[4.4.3. Objeto de control JButton 12](#_Toc104739272)

[4.4.4. Propiedades más usadas: 12](#_Toc104739273)

[4.4.5. Evento más usado: 12](#_Toc104739274)

[5. OPCIONES DEL PROGRAMA 13](#_Toc104739275)

[5.1. Pasos para ejecutar el programa 13](#_Toc104739276)

[5.2. Pasos para el registro de un Paciente: 13](#_Toc104739277)

[5.3. Pasos para actualizar o modificar datos de un Paciente o de su cita 16](#_Toc104739278)

[5.4. Pasos para la eliminar de cita de un Paciente: 18](#_Toc104739279)

[5.5. Pasos para consultar el registro general de citas de los pacientes: 20](#_Toc104739280)

[5.6. Pasos para Limpiar Datos de la interfaz principal: 21](#_Toc104739281)

# ASPECTOS GENERALES

## DEFINICIÓN DEL PROBLEMA:

La problemática en este caso es una demora en reservación y atención de citas en la clínica “LAS AMERICAS”, debido a que esta no cuenta con un sistema web para gestionar de manera ordenada y eficiente una cita, puesto que se realiza de forma manual, y debido a eso se puede provocar una pérdida de información produciendo un desorden en el centro de salud.

Es por ello que es importante un software optimo que permita un orden en la reserva de citas y evitar así inconvenientes en la atención.

### DESCRIPCIÓN DEL PROBLEMA:

Actualmente en algunos hospitales, centros de salud e inclusive clínicas el proceso de reserva de citas es manual, por lo cual pueden presentarse muchos errores.

Esto puede desencadenar diferentes pérdidas, pero sobre todo de clientes, ya que debido a un error de las reservas de citas generaría cruces con las citas de otros pacientes, con los horarios de los médicos, esto produciría reclamos hacia la clínica por retrasos, pérdida de tiempo y deteriorar la imagen de la clínica.

### FORMULACIÓN DEL PROBLEMA

¿De qué manera influye un programa para la gestión de reserva de citas médicas para la clínica?

## DEFINICIÓN DE OBJETIVOS

En vista de la problemática antes mencionada se plantean distintos objetivos que serán mencionados a continuación detalladamente para que se pueda comprender el objetivo de este proyecto.

### OBJETIVOS GENERALES

Desarrollar un Sistema para la gestión de reserva de citas médicas para la clínica.

### OBJETIVOS ESPECÍFICOS

* Implementar un aplicativo intuitivo y accesible, que permita mejorar la gestión de citas médicas para la clínica.
* Disminuir el tiempo de agendamiento de citas médicas desde el acceso de los pacientes, obteniendo como resultados nivel de calidad de atención.
* Gestionar los horarios y turnos de trabajo de los médicos para disponer de un control de asistencia.
* Disponer de un aplicativo que permita realizar citas médicas desde cualquier plataforma a cualquier hora.

# MARCO TEORICO

## Fundamento teórico

El proyecto que se realizará en esta oportunidad consiste en ayudar a nuestros clientes en la solicitud de una cita médica utilizando la plataforma cuyo registro se realizará desde la comodidad de su casa u otro lugar en el menor tiempo posible. La aplicación trabajará internamente mediante un sistema, el cual estará comunicada con una base de datos, quien nos brindará el detalle de la cita generada u otros datos adicionales.

## Planificación del trabajo

Al principio, nos quedábamos una hora aproximadamente en poder debatir y hacer una lluvia de ideas para realizar nuestro proyecto, comenzando en lo que se basaría este mismo. Después de decidirlo, comenzamos creando una base sencilla de cómo se vería la interfaz del proyecto para luego codificarlo. Primero, creamos un nuevo proyecto en el NetBeans, programa utilizado en clase, donde decidimos realizarlo con un jframe; después de eso lo modelamos siguiendo nuestra base, para luego comenzar a codificar uno a uno cada botón del modelado, para que cada uno realice una tarea especifica. Por otro lado, también usamos MySQL, para crear una base de datos y luego unirla a nuestro java. Codificado y funcional todo, solo fue cuestión de adornarlo para que tenga un aspecto más profesional.

## PRODUCTO OBTENIDO

Como resultado final del proyecto creado, se obtuvo la respuesta esperada con cada botón funcional y una base de datos actualizable para poder almacenar información de las citas de la clínica.



# 3. ANÁLISIS DE REQUISITOS

## 3.1. Visión General

Primero que todo debemos tener en cuenta que la aplicación va a ser administrada desde el propio centro clínico en el cual sea instalado, por lo que su administración debe ser sencilla e intuitiva, y hallarse dentro de la misma aplicación, debido a que los usuarios finales son totalmente heterogéneos, reincidiendo así en la usabilidad de la interfaz, la cual debe ser fácil y clara de usar. En cuanto a las funciones de reservación que realizara, serás administradas por la misma aplicación , de introducción de datos y lectura de estos, por lo que la mayor parte de las funcionalidades tendrán que acceder a la base de datos para resolverse.

## 3.2. Requisitos funcionales

En la siguiente lista se dan los requisitos fundamentales que se requieren para la aplicación, según el tipo de usuario que necesitara dichas funcionalidades:

Administrador:

* Crear un nuevo usuario: Crea uno o más usuarios de tipo Recepcionista.
* Borrar un usuario: Elimina una cuenta de usuario de la base de datos, excepto la del mismo Administrador.

Recepcionista

* Registrar datos: Puede registrar datos de los pacientes que quieran reservar una cita.
* Consulta disponibilidad: Puede consultar a la base de datos la disponibilidad de cada especialista.
* Agendar y eliminar cita: Una vez registrado los datos, puede agendar una cita, y si lo requiere la puede eliminar.
* Registro de citas: Puede consultar a la base de datos y ver las citas asignadas a cada horario con su respectivo especialista.

## 3.3. Requisitos de la interfaz de usuario

### 3.3.1. Identificación de las restricciones técnicas

Hay que tener en cuenta que este aplicativo ha de ser usado por el área de recepción de la clínica asignada. Cabe recalcar que funciona de la mano con una base de datos en SQL, lo cual se podía optimizar subiendo un respaldo de en el Internet que funcione de la mano con el aplicativo.

### 3.3.2. Perfiles de usuario

Es necesario hacer una distinción entre todos los tipos de usuarios existentes que usarán la aplicación.

Administrador: Solo existe una cuenta de usuario administrador y es creada en el momento de instalación de la aplicación. Es el encargado de crear el resto de los usuarios que tengan que usar la aplicación.

Recepcionista: Solo existe una cuenta de usuario recepcionista y creada al momento de instalar la aplicación. Es el encargado de organizar y reservar las citas que lleguen en la clínica.

### 3.3.3. Documentación de las tareas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tarea | Frecuencia  (1-4) | Usuarios | Entradas | Salidas | Posibles Problemas |
| Login | 3 | Todos | Username y Contraseña | Devuelve una autentificación de usuario | La entrada no coincide con la BD |
| Cambiar contraseña | 2 | Todos | Contraseña antigua y contraseña nueva | --- | --- |
| Crear nuevo usuario | 3 | Administrador | Datos del nuevo usuario | --- | Usuario ya había sido dado de alta |
| Borrar usuario | 1 | Administrador | Código del usuario | --- | --- |
| Registrar datos |  | Recepcionista | Datos del cliente | Devuelve una autentificación de registro | Datos erróneos |
| Consulta disponibilidad |  | Recepcionista | Consulta a la BD | Disponibilidad del especialista | - Cruce de horarios  - Falla en el sistema |
| Agendar y eliminar cita |  | Recepcionista | DNI del cliente | - Registrado en la BD  - Removido de la BD | - Cruce de horarios  -Falla del sistema |
| Registro de citas |  | Recepcionista | Consulta a la BD | Citas asignadas a los horarios correspondientes | -Falla del sistema |

# TECNOLOGIAS EMPLEADAS:

## NetBeans

NetBeans es un programa que sirve como IDE (un entorno de desarrollo integrado) que nos permite programar en diversos lenguajes.

El desarrollo de software se ha diversificado mucho basándonos en la cantidad de lenguajes que existen para la programación. Sin embargo, hay lenguajes que van imponiéndose como estándares, entre ellos tenemos a Java, PHP, HTML, C++, C#, Ruby.

El problema que se presenta a la mayoría de los programadores es contar con un entorno de desarrollo que sea completo, eficaz, fácil de usar y sea en lo posible gratuito. Todos esos requerimientos los podemos encontrar en NetBeans.

NetBeans es ideal para trabajar con el lenguaje de desarrollo JAVA (y todos sus derivados), así como también nos ofrece un excelente entorno para programar en PHP. También se puede descargar una vez instalado NetBeans, los complementos para programar en C++. La IDE de NetBeans es perfecta. Tiene un excelente balance entre una interfaz con múltiples opciones y el editor puede autocompletar nuestro código.

<https://sites.google.com/site/portafolionetbeans/que-es-netbeans>

## Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de Lenguaje de Programación C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrollados por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

<https://www.ecured.cu/Java_(lenguaje_de_programaci%C3%B3n)>

## MySQL

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto. Desarrollado originalmente por MySQL AB, fue adquirida por Sun MicroSystems en 2008 y esta su vez comprada por Oracle Corporation en 2010, la cual ya era dueña de un motor propio InnoDB para MySQL.

MySQL es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte, es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle.

Las versiones Enterprise, diseñadas para aquellas empresas que quieran incorporarlo en productos privativos, incluyen productos o servicios adicionales tales como herramientas de monitorización y asistencia técnica oficial.

### Características de MySQL

MySQL presenta algunas ventajas que lo hacen muy interesante para los desarrolladores. La más evidente es que trabaja con bases de datos relacionales, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente.

Al ser basada en código abierto es fácilmente accesible y la inmensa mayoría de programadores que trabajan en desarrollo web han pasado usar MySQL en alguno de sus proyectos porque al estar ampliamente extendido cuenta además con una ingente comunidad que ofrece soporte a otros usuarios. Pero estas no son las únicas características como veremos a continuación:

Arquitectura Cliente y Servidor: MySQL basa su funcionamiento en un modelo cliente y servidor. Es decir, clientes y servidores se comunican entre sí de manera diferenciada para un mejor rendimiento. Cada cliente puede hacer consultas a través del sistema de registro para obtener datos, modificarlos, guardar estos cambios o establecer nuevas tablas de registros, por ejemplo.

Compatibilidad con SQL: SQL es un lenguaje generalizado dentro de la industria. Al ser un estándar MySQL ofrece plena compatibilidad por lo que si has trabajado en otro motor de bases de datos no tendrás problemas en migrar a MySQL.

Vistas: Desde la versión 5.0 de MySQL se ofrece compatibilidad para poder configurar vistas personalizadas del mismo modo que podemos hacerlo en otras bases de datos SQL. En bases de datos de gran tamaño las vistas se hacen un recurso imprescindible.

Procedimientos almacenados. MySQL posee la característica de no procesar las tablas directamente, sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación.

Desencadenantes. MySQL permite además poder automatizar ciertas tareas dentro de nuestra base de datos. En el momento que se produce un evento otro es lanzado para actualizar registros o optimizar su funcionalidad.

Transacciones. Una transacción representa la actuación de diversas operaciones en la base de datos como un dispositivo. El sistema de base de registros avala que todos los procedimientos se establezcan correctamente o ninguna de ellas. En caso por ejemplo de una falla de energía, cuando el monitor falla u ocurre algún otro inconveniente, el sistema opta por preservar la integridad de la base de datos resguardando la información.

<https://openwebinars.net/blog/que-es-mysql/>

## JFrame

JFrame es un contenedor de nivel superior que proporciona una ventana en la pantalla. Un marco es en realidad una ventana base en la que se basan otros componentes, a saber, la barra de menú, paneles, etiquetas, campos de texto, botones, etc.

JFrame es una clase de Java que se extiende por la clase Frame de Java. JFrame se trata como la ventana principal. En JFrame se pueden agregar diferentes elementos como etiquetas, campos de texto, botones. Estos elementos en JFrame crean una interfaz gráfica de usuario.

### Objeto de control JTextField

Un objeto de control JTextField permite dibujar en el formulario un cuadro de texto, es

decir, una caja que permite la introducción de un dato o valor. Este objeto es utilizado para

el ingreso de datos.

### Métodos más usados:

getText(): Permite obtener el texto introducido en el cuadro de texto.

setText(): Permite colocar un texto en el objeto JTextField.

JButton(): Permite crear un botón etiquetado que tiene una implementación independiente de la plataforma.

### Objeto de control JButton

Un objeto de control JButton permite dibujar en el formulario un objeto que contiene un

proceso para ejecutar. Se utiliza comúnmente para llevar a cabo procesos específicos según

la naturaleza de la aplicación.

## Propiedades más usadas:

● Text: Contiene el valor o dato introducido en el cuadro de texto.

## Evento más usado:

● ActionPerformed: Este evento se lleva a cabo cuando el usuario da click sobre el

objeto de control JButton.

# OPCIONES DEL PROGRAMA

## Pasos para ejecutar el programa

Click derecho sobre la clase Frame con el método Main llamada Pacientes.java y seleccionar la opción Run File.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Pasos para el registro de un Paciente:

* **Paso nº1:** Ingresar datos del cliente como Nombres, DNI, teléfono.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

* **Paso nº2:** Seleccionar la especialidad y presionar el botón seleccionar especialidad, automáticamente figuraran los doctores disponibles.

Interfaz de usuario gráfica

Descripción generada automáticamente

* **Interfaz de usuario gráfica

  Descripción generada automáticamentePaso nº3:** Seleccionar un doctor y presionar el botón Procesar Doctor, automáticamente aparecerán los horarios disponibles del médico.
* **Paso nº4:** Seleccionar el horario y darle al botón Registrar Cita, automáticamente nos votara el siguiente mensaje. Presionaremos Ok

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

* **Paso nº5:** Para verificar el correcto registro nos iremos a el botón consultar registros y automáticamente nos llevara a una tabla con todos los pacientes registrados.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

## Pasos para actualizar o modificar datos de un Paciente o de su cita

* **Interfaz de usuario gráfica, Sitio web

  Descripción generada automáticamentePaso nº1:** En la interfaz Principal, seleccionar el botón actualizar datos
* **Paso nº2:** Seguir las instrucciones del aviso y Hacer click en consultar Cita, aparecerá la siguiente ventana donde deberán ingresar el nombre del paciente y darle a ok.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

* **Paso nº3:** Les aparecerá automáticamente los datos del paciente registrado, ahora solo deben cambiar el dato que deseen modificar

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

* **Paso nº4:** Una vez modificado los datos que desee cambiar presionar el botón actualizar Datos y automáticamente aparecerá el siguiente mensaje.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

* **Texto

  Descripción generada automáticamente con confianza mediaPaso nº5:** Para verificar la correcta actualización nos iremos a el botón consultar registros y automáticamente nos llevara a una tabla con todos los pacientes registrados.

## Pasos para la eliminar de cita de un Paciente:

* **Paso nº1**: En la interfaz principal, seleccionar el botón Eliminar Cita, automáticamente nos llevara a nuestra tabla de pacientes registrados

Interfaz de usuario gráfica

Descripción generada automáticamente

* **Paso nº2:** En tabla seleccionar la fila con los datos del paciente a eliminar

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

* **Paso nº3:** Darle click al botón eliminar y automáticamente nos botara este mensaje

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* **Paso nº4:** Para verificar que se eliminó la cita del paciente simplemente presione Ok en el mensaje y la tabla se actualizara

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Pasos para consultar el registro general de citas de los pacientes:

* **Paso nº1:** En la interfaz principal, seleccione el botón consultar registros

Interfaz de usuario gráfica

Descripción generada automáticamente

* **Texto

  Descripción generada automáticamente con confianza mediaPaso nº2:** Se abrirá automáticamente la tabla de registros generales, podrá visualizar el dato o datos que desee
* **Interfaz de usuario gráfica, Aplicación

  Descripción generada automáticamentePaso nº3:** Para volver a la interfaz principal seleccione Archivo en la parte superior izquierda de la ventana de Consulta de registro de pacientes y seleccione Registro de Citas o en su defecto presione Ctrl+T
* **Paso nº4:** Ya está de vuelta en la interfaz principal

Interfaz de usuario gráfica

Descripción generada automáticamente

## Pasos para Limpiar Datos de la interfaz principal:

* **Paso nº1:** Suponiendo que tiene la tabla llena de algún dato o muchos datos se mostraría algo así

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

* **Paso nº2:** Presione el botón Limpiar datos y automáticamente todos los campos se vaciarán sin excepción alguna.

Interfaz de usuario gráfica

Descripción generada automáticamente