

## **“CONSULTORIO DENTAL”**

### **ING. SISTEMAS COMPUTACIONALES**

PRESENTA:

**ÁLVAREZ VILLAGOMEZ KARENINA VERENICE**

**BALCAZAR ORTIZ MICHELLE ARILENI**

**CERRITOS CAMARGO CRISTIAN OMAR**

**CHÁVEZ SÁMANO PEDRO ANTONIO**

**FRANCO ORTIZ DANIEL FRANCISCO**

DOCENTE:

**VEGA CHÁVEZ EFREN**

**Uriangato, Gto.**

**ENERO, 2017.**



## Tabla de contenido

<b>Introducción .....</b>	<b>6</b>
Propósito.....	6
Ámbito de sistema .....	6
Definiciones, acrónimos y abreviaturas .....	6
Referencias.....	6
<b>Capítulo I. Antecedentes .....</b>	<b>7</b>
1.1 Información de la empresa.....	7
1.2 Planteamiento del problema .....	7
1.3 Justificación .....	7
1.4 Objetivos .....	7
1.4.1 Objetivo general.....	7
1.4.2 Objetivo específico.....	8
<b>Capítulo II: Fundamento teórico.....</b>	<b>8</b>
2.1 Software a la medida.....	8
2.2 Gestores de bases de datos.....	9
2.3 Lenguajes de programación.....	9
2.4 Ing. de software.....	9
<b>Capítulo III: Metodología .....</b>	<b>10</b>
3.1 Análisis de requerimientos .....	10
3.2 Características de los usuarios .....	10
3.3 Restricciones .....	10
3.4 Suposiciones y dependencias .....	14
<b>Capítulo IV: Resultados.....</b>	<b>17</b>
<b>Conclusiones.....</b>	<b>15</b>
<b>Competencias desarrolladas y aplicadas.....</b>	<b>16</b>

<b>Bibliografía.....</b>	<b>16</b>
--------------------------	-----------

## Índice de Tablas.

Tabla 1. Definiciones, Acrónimos y abreviaturas.....	6
Tabla 2. Referencias.....	6
Tabla 3. Características de los usuarios.....	7

## Índice de figuras.

Figura 1. Diagramas Relacionales. ....	<b>¡Error! Marcador no definido.</b>	<b>3</b>
Figura 2. Conexión de Base de Datos.....		13
Figura 3. Mensajes de Error.....		13
Figura 4. daoCitas.....		14
Figura 5. daoPacientes.....		14
Figura 6. daoPagos.....		15
Figura 7. daoProcdutos.....		15
Figura 8. Interfaz Agregar Pacientes.....		16
Figura 9. Interfaz Catálogo de Pacientes.....		16
Figura 10. Interfaz Final.....		17

## INTRODUCCION

Este documento es una Especificación de Requisitos Software (ERS) para el Sistema de información para la gestión de procesos y control de inventarios. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE.

### PROPOSITO.

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para el desarrollo de un sistema que permitirá gestionar distintos procesos administrativos, mejorando el rendimiento y el trato de los odontólogos y haciendo la visita de los pacientes más cómoda y de una manera más segura, con forme al seguimiento de sus tratamientos y datos personales.

### AMBITO DEL SISTEMA.

El sistema aún no cuenta con un nombre específico, pero si los beneficios que ofrecerá tanto al cliente como al profesional (usuario). El sistema tendrá una sección especial para revisar el avance de los pacientes, se podrán revisar los videos o placas del mismo para ver si el tratamiento está rindiendo frutos.

### DEFINICIONES, ACRONIMOS Y ABREVIATURAS.

<i>Nombre</i>	<i>Descripción</i>
<b>Usuario</b>	Persona que usará el sistema para gestionar procesos
<b>ERS</b>	Especificación de Requisitos Software
<b>RF</b>	Requerimiento Funcional
<b>RNF</b>	Requerimiento No Funcional
<b>FTP</b>	Protocolo de Transferencia de Archivos

Tabla 1. Definiciones, Acrónimos y abreviaturas.

### REFERENCIAS

<b>Título del Documento</b>	<b>Referencia</b>
Standard IEEE 830 - 1998	IEEE

Tabla 2. Referencias

## **CAPITULO I. ANTECEDENTES.**

### **1.1 Información de la empresa.**

El consultorio Dental e la Dra. Anel Gómez Meraz es muy reconocido en la región porque cuneta con múltiples servicios, accesorios, tratamientos, entre otras cosas. Por esta razón existe la posibilidad de incrementar las ganancias que se obtienen, si el proceso de los registros es más sencillo y eficaz, además de que el trato a sus pacientes es de mejor calidad. Por eso es que se llegó a la idea de crear un software para mejorar los servicios de dicho consultorio (administración de pacientes, productos, creación de una nueva agenda, historiales médicos).

### **1.2 Planteamiento del problema.**

El consultorio es un establecimiento que cuenta con una amplia variedad de servicios, pero la administración de datos de los pacientes, productos, entre otras cosas, cuentan con un registro a base de papel y lápiz, lo cual dificulta estar buscando el historial de cada paciente que llega al consultorio. Por lo que se desperdicia tiempo y el trato al paciente no es de lo más eficaz, lo que se podría ahorrar y dar mejor trato al contar con un software que brinde un forma de registro aún más rápida y de buena calidad.

### **1.3 Justificación.**

Hoy en día es muy importante tener nuevas tecnologías que nos ayuden a realizar tareas cotidianas, porque eso nos resuelve muchos problemas que de otra forma podrían llegar a ser imposibles.

### **1.4 Objetivos**

#### **1.4.1 Objetivo general.**

Realizar un software que ayude a la buena administración y captura de datos de los pacientes.

#### 1.4.2 Objetivos específicos.

- Analizar los requerimientos del cliente.
- Aplicar patrones de diseños modernos.
- Implementar la solución de las necesidades del cliente.
- Comprobar el funcionamiento de cada método.
- Instalar el software.

## **CAPITULO II. FUNDAMENTOS TEORICOS.**

### **2.1 SOFTWARE A LA MEDIDA.**

Software a la medida es aquel sistema que se diseña y desarrolla de manera personalizada y única. Es decir, busca complacer todas las necesidades y adaptarse lo mejor posible a lo que una empresa necesita.

Objetivos:

Algunos de los objetivos más importantes que se planean implementar durante el ciclo de vida del software son los siguientes:

- Reducir tiempos en la realización de inventarios.
- Tener un registro apropiado de los pacientes.
- Llevar un control de inventario de productos.

Análisis de los requisitos y su viabilidad

- Se podrá registrar, modificar y eliminar pacientes en la base de datos.
- Se podrá insertar, modificar y eliminar productos en el inventario.
- Llevará un control de la contabilidad.

Diseño general

- El software cuenta con una pantalla principal en la cual te direcciona a las demás interfaces.
- Cuenta con una interfaz para los pacientes y otra para el inventario.
- También se cuenta con una base de datos en la que se almacenará toda la información.



### Programación

1. Cada clase deberá llamarse según el código programando, por ejemplo, una clase se llamará DatosActividad y en esta se incorporará el código correspondiente para hacer el llenado de la base de datos en la tabla de actividades.
2. También al principio de cada clase se deberá comentar que es lo que se realizará en las líneas de código siguientes y cada método deberá estar comentado para saber qué es lo que realiza cada uno.
3. para la creación de las variables, el nombre de estas deberá estar relacionado con lo que almacenará cada uno de los datos en la base de datos.
4. Las variables se deberán declarar al principio del método o de la clase donde serán usadas

### MANTENIMIENTO

- Se prevé que el sistema necesitará ser actualizado a un año de su implementación.

## **2.2 GESTORES DE BASES DE DATOS**

El gestor de la base de datos que utilizamos fue MySQL workbench.

## **2.3 LENGUAJE DE PROGRAMACION.**

Optamos por usar el lenguaje de programación C# ya que es el lenguaje que manejamos con más eficiencia.

## **2.4 INGENIERIA DE SOFTWARE.**

La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos (software).

## CAPITULO III.- METODOLOGÍA

### 3.1 ANÁLISIS DE REQUERIMIENTOS.

Investigar los requerimientos del cliente en esta fase se realizaron encuesta o entrevistas como se muestran a continuación.

¿Qué servicios quiere que tenga su software?

¿Tiene algún patrón de diseño en su mente para el desarrollo del software?

¿Tiene alguna fecha en que desea que su software este terminado?

¿Desea generar facturas?

### 3.2 DISEÑAR EL SOFTWARE

En esta fase se realizaron

#### 1. Diagrama relacional(BD)

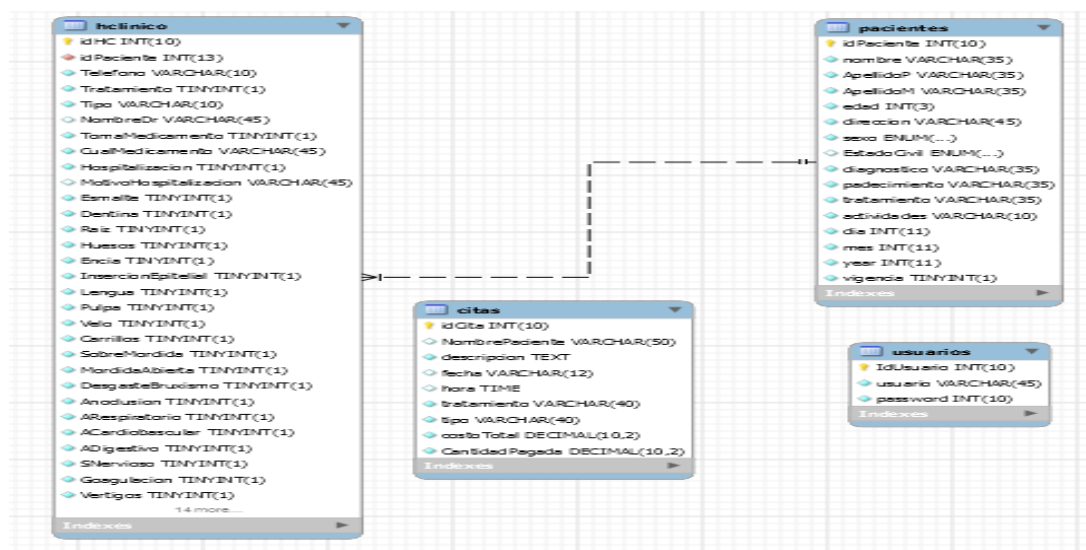


Fig. 1. Diagramas Relacionales.

### 3.3 Implementar la solución de las necesidades del cliente.

En este segmento de código se está haciendo la conexión a la base de datos, y en caso de que este falle se muestra un mensaje indicando este error.

```

public class Conexion {
    private static Connection conexion;
    private static Statement sentencia;

    public static void conecta() {
        try {
            conexion = DriverManager.getConnection("jdbc:mysql://localhost/consultorio?user=root&password=root");
            sentencia = conexion.createStatement();
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null,
                "Ha ocurrido un error al establecer la conexion",
                "Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    public static ResultSet ejecutarSQLSelect(String sql) {
        try {
            sentencia = conexion.createStatement();
            return sentencia.executeQuery(sql);
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
            System.out.println(ex.getMessage());
            return null;
        }
    }
}

```

Fig. 2. Conexión de Base de Datos.

```

public static boolean ejecutarSQL(String sql) {
    try {
        sentencia = conexion.createStatement();
        sentencia.executeUpdate(sql);
        return true;
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
        System.out.println(ex.getMessage());
        return false;
    }
}

public static void cerrarConexion() {
    try {
        conexion.close();
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
        System.out.println(ex.getMessage());
    }
}
}

```

Fig. 3. Mensajes de Error.

En este segmento de código daoCitas se hace la función insertar que nos ayuda a agregar datos de las citas, tales como ID cita, nombre del paciente, descripción, fecha, hora, tratamiento, tipo de pago, costo total, cantidad pagada.

```

public class daoCitas implements IDao<Citas> {

    @Override
    public boolean Insert(Citas obj) {
        String sql = "insert into citas values(" + obj.getIdCita() + "," +
            obj.getNombrePaciente() + "','" + obj.getDescripcion() + "','" +
            obj.getFecha() + "','" + obj.getHora() + "','" + obj.getTratamiento() + "','" +
            obj.getTipo() + "','" + obj.getCostoTotal() + "','" + obj.getCantidadPagada() + ")";
        return Conexion.ejecutarSQL(sql);
    }
}

```

Fig. 4. daoCitas.

En este segmento de código daPacientes se hace la función insertar que nos ayuda a agregar datos de los pacientes, tales como ID Paciente, nombre del paciente, apellidos, sexo, edad, estado civil, tratamiento, actividades, padecimiento.

```

-  */
public class daoPacientes implements IDao<Pacientes> {

    @Override
    public boolean Insert(Pacientes obj) {
        String sql = "insert into pacientes values(" + obj.getIdPaciente() + "," + obj.getNombre()
            + "','" + obj.getApellidoP()
            + "','" + obj.getApellidoM() + "','" + obj.getEdad() + "','" + obj.getDireccion()
            + "','" + obj.getSexo() + "','" + obj.getEstadoCivil() + "','" + obj.getDiagnostico()
            + "','" + obj.getPadecimiento() + "','" + obj.getTratamiento() + "','" + obj.getActividades()
            + "','" + obj.getDia() + "','" + obj.getMes() + "','" + obj.getYear() + "','" + obj.isVigencia() + ")";
        return Conexion.ejecutarSQL(sql);
    }
}

```

Fig. 5. daoPacientes.

En este segmento de código daoPagos se hace la función insertar que nos ayuda a agregar datos de los pagos, tales como ID producto, ID paciente, descripción, Cantidad, pago, deuda.

```

public class daoPagos implements IDao<pojos.Pagos> {

    @Override
    public boolean Insert(Pagos obj) {
        String SQL = "Insert into pagos values (null," + obj.getIdProducto() + "," + obj.getIdPaciente()
            + "','" + obj.getDescripcion() + "','" + obj.getCantidad() + "','" + obj.getPago() + "','" + obj.getDebe() + ")";
        return Conexion.ejecutarSQL(SQL);
    }
}

```

Fig. 6. daoPagos.

En este segmento de código daoProductos se hace la función insertar que nos ayuda a agregar datos de los productos, tales como ID producto, nombre del producto, tipo, Cantidad, precio de compra, precio de venta.

```
public class daoProductos implements IDao<Productos> {

    @Override
    public boolean Insert(Productos obj) {
        String SQL = "insert into productos values (" + obj.getIdProducto() + "," +
            + obj.getNombre() + "," + obj.getTipo() + "," + obj.getDescripcion() + "," +
            + obj.getCantidad() + "," + obj.getPrecioCompra() + "," + obj.getPrecioVenta() + ")";
        return Conexion.ejecutarSQL(SQL);
    }
}
```

Fig. 7. daoProductos.

### 3.4 Comprobar el funcionamiento de cada Meodo.

En el método insertar información se solicita a los clientes id paciente, nombre, apellido paterno, apellido materno, dirección, diagnostico, fecha de nac, edad, sexo, estado civil, padecimiento, plan de tratamiento, actividades, frecuente o no frecuente.

Fig. 8. Interfaz Agregar Pacientes.

Aquí se muestran los registros que se van ir insertando.

idPaciente	nombre	ApellidoP	ApellidoM	edad	direccion	sexo	EstadoCivil	diagnostico	padecimiento
1	Pedro Antonio	Chavez	Samano	21	Michoacan #7	Masculino	Union Libre	Ninguno	Ninguno
2	Yuli	Hernandez	Pantoja	21	Av. Puebla #45	Femenino	Union Libre	Ninguno	No
3	Nelly	Gomez	Meraz	20	Hemosillo #28	Femenino	Soltero(a)	No	Ninguno
4	Daniel	Franco	Ortiz	20	Av. Morelos	Masculino	Soltero(a)	No	No
5	Jonathan	Camacho	Lopez	20	Itsur	Masculino	Divorciado(a)	Medio Loco	Ninguno
6	Omar	Centos	Camargo	20	JoseMateo #45	Masculino	Soltero(a)	Ninguno	Ninguno

Fig. 9. Interfaz Catálogo de Pacientes.

## CAPÍTULO IV. RESULTADOS.

Ya con el sistema implementado dentro del consultorio odontológico, se llegaron a los siguientes resultados:

- Se mejorará el formato de registro de pacientes en la empresa.
- Se reducirá el tiempo de venta de productos al cliente.
- Se generarán registros confiables de la producción de ventas.
- Se garantizará que un mejor servicio al cliente.
- Se mejorara el tiempo de registro de las citas y se tendrá mejor organización para la Agenda.



Fig. 10. Interfaz Final.

## Conclusiones

Aprendimos a establecer los diferentes casos de uso, buscar sus diferentes requerimientos funcionales a través de distintas técnicas como pueden ser entrevistas, lluvia de ideas y sobre todo, saber bien en donde se estará trabajando, ya que al conocer las instalación de la empresa (consultorio odontológico) se puede observar mejor cada aspecto, esta nos permitió saber lo que necesitaba la empresa en su proyecto para solucionar problemas.

Además, planeamos todo el funcionamiento del software desde la arquitectura de clases, los diagramas de secuencia y modelos de negocios que mostraban paso a paso como el usuario ingresaría al sistema, realizaría lo que deseaba y terminaría el uso del sistema.

## Competencias desarrolladas y aplicadas

Más que otra cosa entre los aprendizajes obtenidos está el saber recaudar la información necesaria e interpretarla para lograr lo que el cliente requería para su empresa. También se implementaron librerías para el estilo de las interfaces de usuario, estas herramientas nos ayudan a darle mejor vista al software.

## BIBLIOGRAFÍA.

Rodriguez, N. D. (2006). *Programacion Orientada a Objetos*.  
Sierra, C. (2012). *Head First Software Development*.