



# Informe final

## Proyecto de gestión de información en una microempresa.

**NOMBRE:** Ricardo Arriagada – Ismael Espinoza – Felipe Sánchez – Claudio Vargas

**CARRERA:** Ing. Informática

**ASIGNATURA:** Sistemas de información

**PROFESOR:** Fabio Duran

**FECHA:** julio 11 de 2017

## Índice

Introducción	3
Plan de trabajo	4
Validación y encuesta	5
Análisis y análisis DAFO	6
Requerimientos y funcionalidades	8
Conclusión	9
Anexos	10

## Introducción

El presente trabajo, contempla el entendimiento de uno de los tantos problemas que puede tener una pequeña o microempresa: la gestión de la información de los clientes. Esto va más allá del rubro en el cual se desenvuelve Winebroker Ltda. empresa dedicada al rubro del corretaje vitivinícola. Este es un problema, que puede presentarse en cualquier rubro por lo cual, ahondar en detallar el rubro del corretaje de vinos, no tiene cabida en este informe.

La motivación para escoger este problema radica en que es una temática que puede darse en cualquier tipo de empresa que cuente con un área administrativa, por lo que se complementa con las áreas de finanzas y recursos humanos.

Cubrir la problemática de la implementación por costos, es lo que nos motiva también a entregar un sistema que sea escalable en crecimiento para así, generar un menor impacto en el presupuesto de la misma empresa. Generar un sistema modular es la solución más factible y práctica para la implementación del software.

Las causas de que exista esta problemática radica en dos premisas, la poca experiencia de los usuarios con sistemas externos a una ofimática arraigada en el uso cotidiano y la poca implementación de nuevas tecnologías dentro de la empresa, por los altos costos y riesgos.

El análisis del problema en sí comienza en la observación de la modalidad de trabajo de las personas que conforman la empresa, la cual a primera vista demuestra una gran dependencia de las herramientas de ofimática, además del uso del almacenamiento compartido gracias al uso de herramientas de almacenamiento en línea en modalidad cloud.

Por parte de nuestro grupo, las tareas asignadas según la rúbrica fueron realizadas a lo largo del semestre, lo cual se ve reflejado en el presente trabajo.

## **Plan De Trabajo**

### **Roles.**

- Ricardo Arriagada: Contacto con la Empresa y coordinador
- Ismael Espinoza: Encargado de programación del prototipo
- Felipe Sánchez: Diseño de diagramas y corrección de detalles en prototipo
- Claudio Vargas: Encargado de la revisión y desarrollo de informe

### **Metodología de trabajo.**

La metodología de trabajo interna del grupo se detalla en los siguientes puntos:

- Comunicación a través de las diversas plataformas de comunicación online (Google Drive, Gmail, WhatsApp, GitHub, etc.).
- Reuniones presenciales en horas de clases para dar acuerdo y constancia de lo que se está trabajando.
- Problemas puntuales fueron solo en dos clases las cuales no nos pudimos reunir todos físicamente en el aula de clases, por enfermedad y ausencia de compañeros.
- Los objetivos son avanzar en cada clase dos o al menos un punto de la actividad, pero nunca dejar nada sin resolver hasta estar todos de acuerdo.
- La problemática se va resolviendo mediante investigación previa y acorde vaya avanzando la actividad.
- La toma de decisiones siempre fue de manera grupal, comunicando cualquier cambio o sugerencia a través de los medios de comunicación que tenemos al alcance.
- Adicionalmente se debe mencionar que existió un trabajo individual de cada integrante del grupo en sus hogares.

### Validación.

Para verificar la existencia del problema y los efectos que podría traer el cambio a un nuevo sistema dentro de la empresa, hemos realizado una encuesta a dos trabajadores de esta, la cual está compuesta de tres preguntas similares a cada uno y de esta manera, obtener una idea de qué tan factible es implementar este software.

#### Desarrollo de la encuesta a trabajadores.

1.- ¿Cree usted que sería una buena idea recurrir a la adquisición de un software para acabar con la desorganización de información dentro de la empresa?

**Trabajador 1:** *creo que sería útil entregar una herramienta distinta al Excel, ya que estaría la información más a mano.*

**Trabajador 2:** *cualquier alternativa para manejar los datos, sería de gran ayuda, ya que, si un trabajador falta, la información va a estar disponible para todos.*

2.- ¿Cree usted que se le sería difícil acostumbrarse al nuevo sistema?

**Trabajador 1:** *puede ser que sea difícil usar algo nuevo, hay que adaptarse a las tecnologías.*

**Trabajador 2:** *todos los días usamos el teléfono y el computador, no creo que vaya a ser muy distinto a eso.*

3.- ¿considera que sería un beneficio a futuro para la empresa la implementación de este sistema?

**Trabajador 1:** *si la idea es que la empresa tenga al mediano plazo un sistema a medida, sería muy bueno.*

**Trabajador 2:** *a futuro, lo ideal sería que fuese más fácil encontrar la información de las personas con las que trabajamos, por lo que estimo que sería de gran utilidad.*

## Análisis

### Problema

Dado los requerimientos de la gestión de información que necesitan las empresas, a día de hoy incorporar un software ERP, puede no ser la mejor opción, ya que son softwares generalizados no adecuados a los requerimientos acotados de por ejemplo las pymes, las cuales, por falta de recursos o por metodologías de gestión, prefieren reevaluar las opciones de crecimiento tecnológico o se realizan malas inversiones.

Es por esto que, observando el entorno de desarrollo de una pequeña empresa que se desenvuelve en el mundo del vino, hemos visto que, existe un problema de gestión de la información que manejan, las cuales agrupan en tablas de formato .xlsx, lo cual hace ineficiente su uso compartido por los integrantes de dicha empresa.

la idea de generar un software de gestión de información a medida el cual principalmente facilitará el uso por parte de los usuarios. La implementación del Software (hay que darle un nombre, para no ser tan genérico) aumentará el rendimiento de los trabajadores, al facilitar el acceso a toda la red de contactos de manera eficiente permitiendo una mayor claridad para identificar cada rol o cargo dentro de una empresa externa.

### Análisis DAFO

- **Debilidades:** implementar un software a medida, implica que los usuarios del mismo, deben aprender algo nuevo, algo a lo que las personas en general no están acostumbradas, y en nuestro caso si vemos que ellos no son capaces de adaptarse al cambio veremos que nuestro software no fue bien recibido por parte de ellos.
- **Fortalezas:** diseñar un software a medida es la mejor solución para el o los problemas específicos que presenta una empresa, lo que asegura un rendimiento óptimo al momento de trabajar con el mismo, al igual que a nosotros se nos haría mucho más fácil y menos engorroso el llevarlo a cabo debido a nuestros conocimientos en el área de desarrollo de software.
- **Amenazas:** Si bien la resistencia al cambio es la principal amenaza, ya que el uso extensivo de la suite de escritorio de Microsoft, hace, de cierto modo, dar cierta confianza a los usuarios de sentir que la herramienta que utilizan es la mejor para desarrollar el trabajo, a su vez también está la amenaza de no poder llevar a cabo el proyecto, que se nos encargó, ya que por cualquier "X" motivo, no lo podamos concretar a tiempo o de la forma que se había planeado.
- **Oportunidades:** la idea de implementar mejoras en los procesos productivos, de entregar un producto a medida con los requerimientos y contemplando también las necesidades de los usuarios finales es la oportunidad a destacar, para entregar una gestión de datos mejorada y más segura.

**Requerimientos:** El problema, ha sido entendido por todos los integrantes del grupo ya que dentro del mismo se han generado algunas propuestas para desarrollar la solución, para ello, se han encontrado las distintas instancias de desarrollo:

Una de las tareas y mejoras que se deberían implementar es un acceso definido para cada usuario (perfiles) como a su vez la expiración de secciones abiertas por estos, otro punto que se tiene en cuenta y en mente para desarrollar en un futuro es la restricción de acceso a la hora de obtener la información; que quiero decir con esto, que como se deja saber en la problemática cualquier usuario puede obtener acceso a la nube y modificar información importante, y es esto más que nada lo que se busca evitar.

**Actores:** Los actores que se presentan en este sistema son dos: Operador y Base De Datos, los cuales interactúan entre si a través de una interfaz “*prototipo o Software de gestión de información*” permitiendo la manipulación de los datos.

**Solución:** La propuesta principal es desarrollar un software para gestionar la información interna de un área de una pequeña empresa dedicada al rubro vitivinícola. La necesidad de implementar una solución global para mejorar la comunicación interna en los distintos niveles del personal que forma parte de la empresa, el acceso ágil y seguro a los datos que se gestionan para la emisión de documentos y ordenamiento.

Costo para la empresa. El software de gestión de información, se presenta como una alternativa real para facilitar el trabajo de las personas que conforman Winebroker. El precio de este software en una primera instancia sería de \$100.000 pesos, además del precio de instalación, costo que la empresa estaría dispuesta a asumir, ya que sería una implementación que contempla como idea futura el desarrollo de módulos adicionales para que, con el pasar del tiempo, confluirá en la implementación plena de un software a medida.

### **Requerimientos y/o Funcionalidades**

#### **Diagramas.**

Los diagramas correspondientes al presente proyecto se encuentran detallados y adjuntos como anexo a este informe, además de encontrarse en el respectivo repositorio. Estos diagramas fueron desarrollados con la herramienta de desarrollo de diagramas UML “DÍA”.

#### **Construcción de prototipo.**

La construcción del prototipo a entregar para solucionar el problema que presenta la empresa, lo hemos denominado *software de gestión de información*, el cual consiste en la ejecución de un CRUD además de un inicio de sesión con usuario y contraseña.

Este *software de gestión de información* fue desarrollado en el lenguaje de programación “C#”, el cual fue escogido por la habilidad con la que cuentan los integrantes de este grupo con dicho lenguaje. Además, es importante destacar que C# permite una facilidad adicional al compatibilizarse bien con herramientas de ofimática incorporadas en el entorno Windows (Office).

Los códigos del prototipo correspondientes al presente proyecto se encuentran detallados y adjuntos como texto plano en el anexo de este informe, además de encontrarse en el respectivo repositorio.



## **Conclusión.**

La implementación de un software dirigido a la gestión de la información de las personas que conforman las distintas empresas con las que trabaja Winebroker, sería de un corto plazo y bajo costo, además la adaptación de los usuarios a esta nueva plataforma sería de un bajo impacto para la productividad de la misma, ya que el software de solución de gestión en sí ayuda a organizar las estructuras organizacionales de los clientes.

El objetivo final es crear un canal, que permita a los trabajadores interactuar de manera ágil, segura y sencilla con la información de los clientes, lo cual con la implementación del software de gestión de información se cumple con esta tarea.

El desarrollo de herramientas que facilitan el acceso a la información es la esencia del trabajo de la creación de software, esperamos en un futuro, continuar con el trabajo continuo con microempresas, ayudando al crecimiento de las mismas, entregando soluciones acotadas a los problemas que estas presentan.

---

## **Anexos**

*Proyecto de gestión de información en una microempresa*

---

## Diagramas.

### Caso de uso

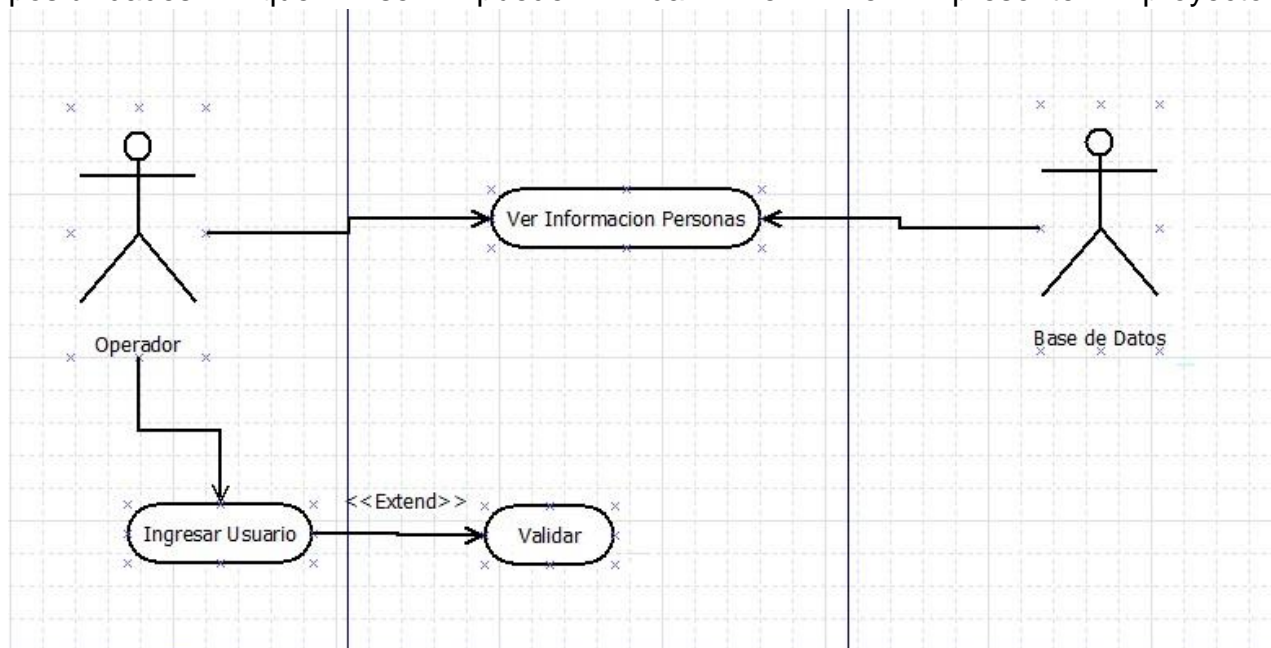
Como se puede apreciar en la figura contamos con un actor en este caso un operador el cual interactúa directamente con una base de datos dentro de la cual puede ejecutar diferentes acciones como, por ejemplo: editar la información.

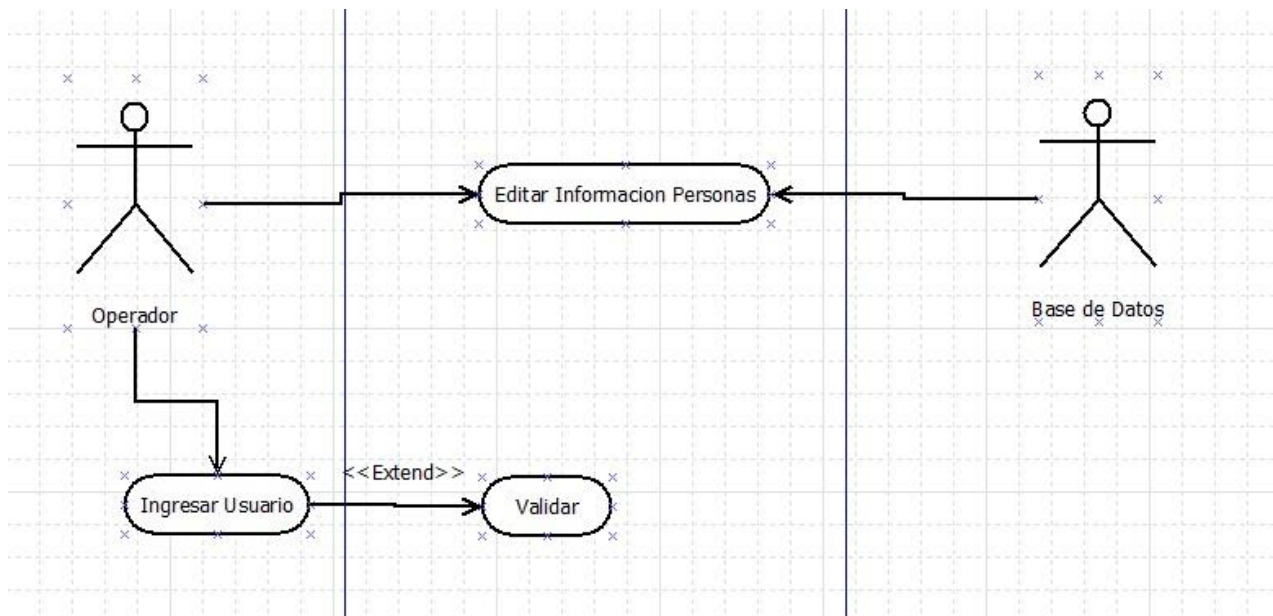
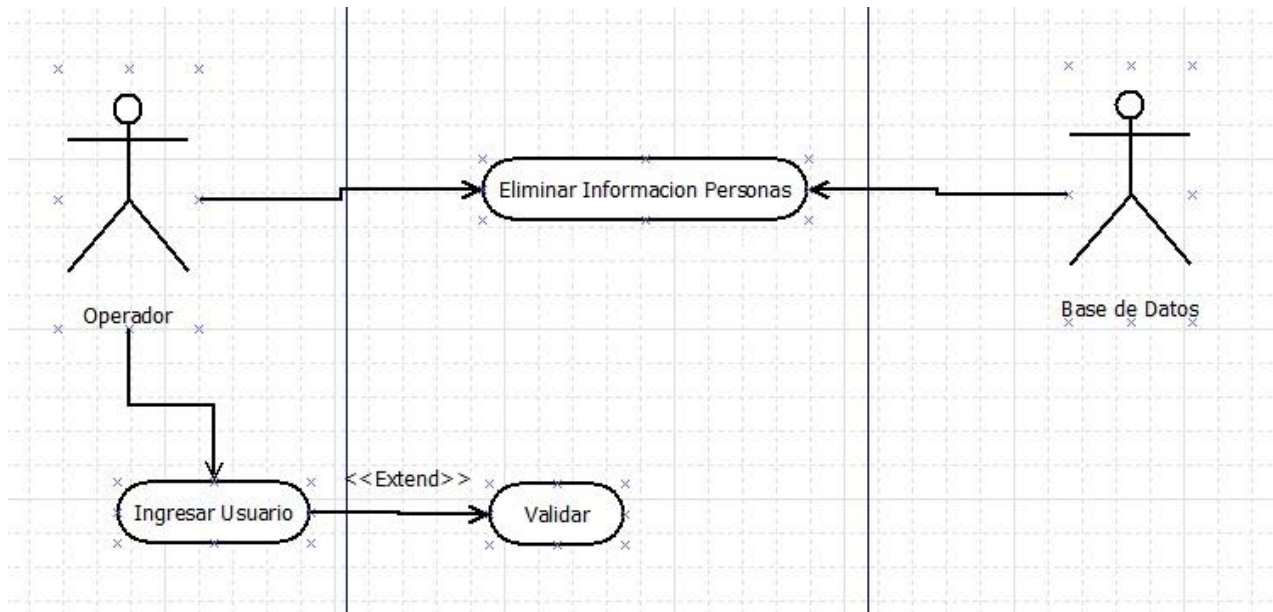
Actores:

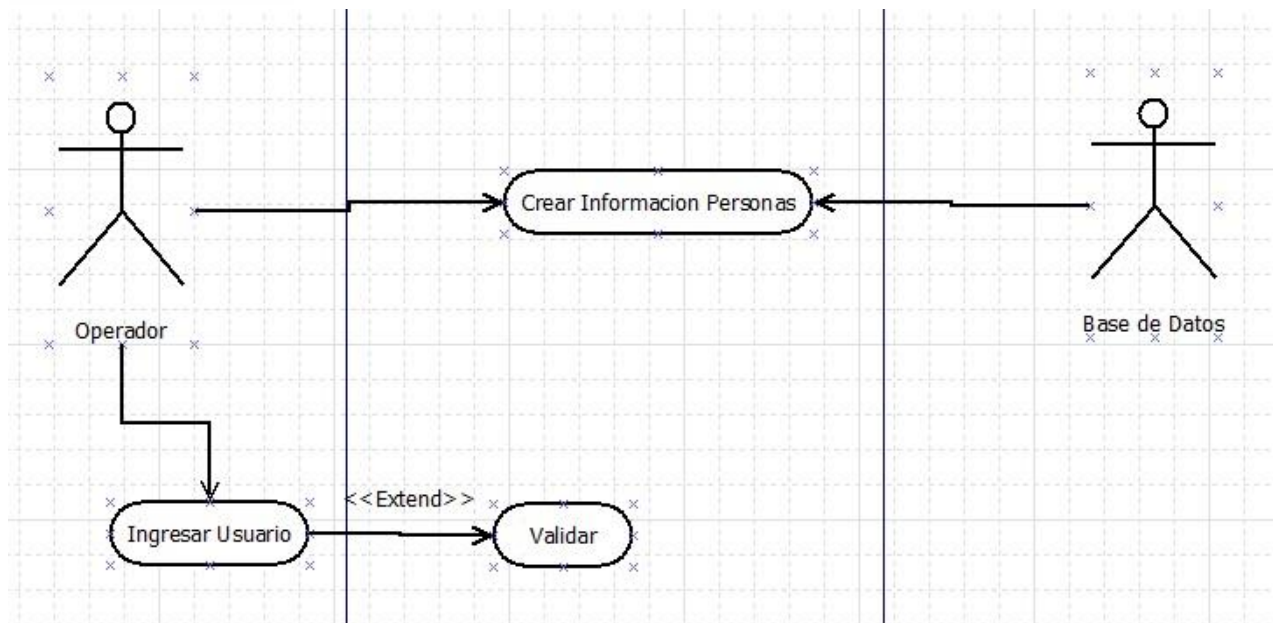
1.-Operador: Usuario que tiene los privilegios de editar, ver, eliminar y crear información dentro de la base de datos

2.-Base De Datos: Actor que tiene toda la información ingresada por el operador y es quien se encarga de enviar, recibir y registrar la información ingresada por el usuario

Los diagramas presentados a continuación se muestran detallando cada una de las posibilidades que se pueden dar en el presente proyecto.

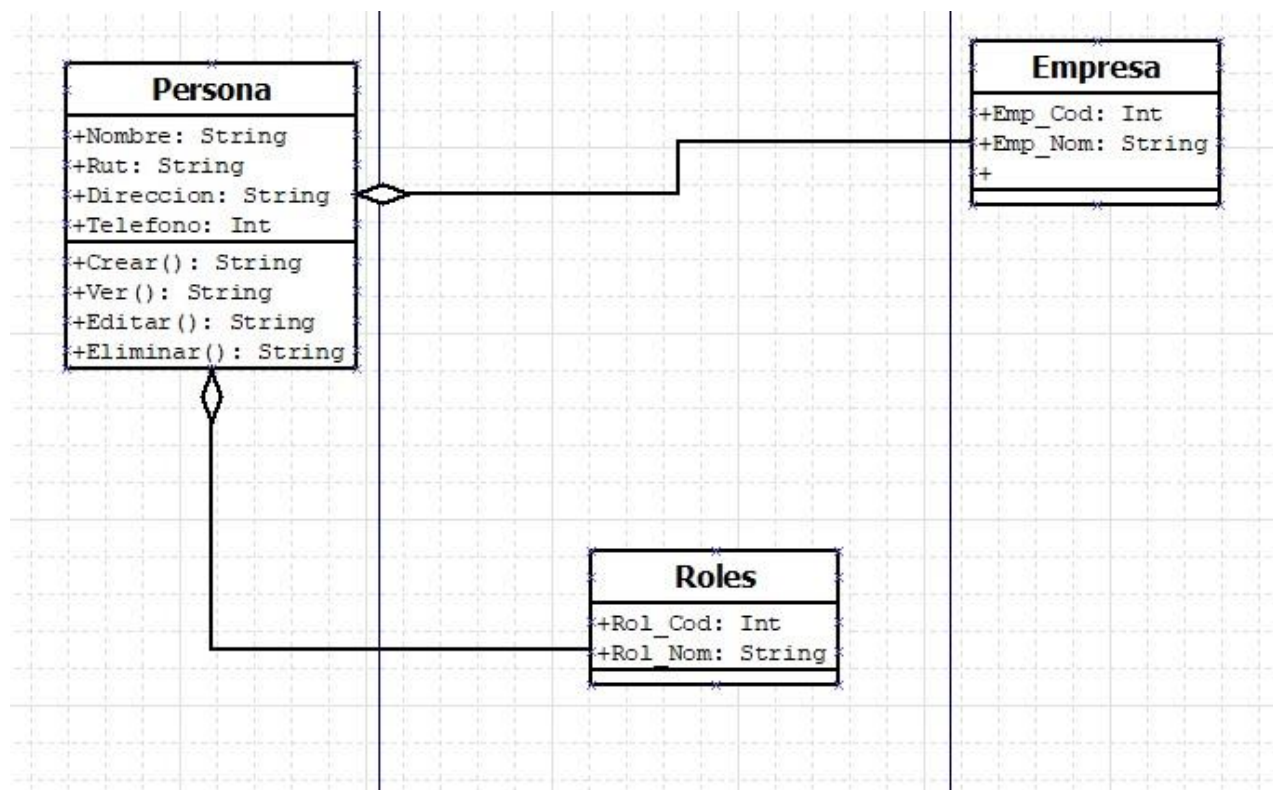






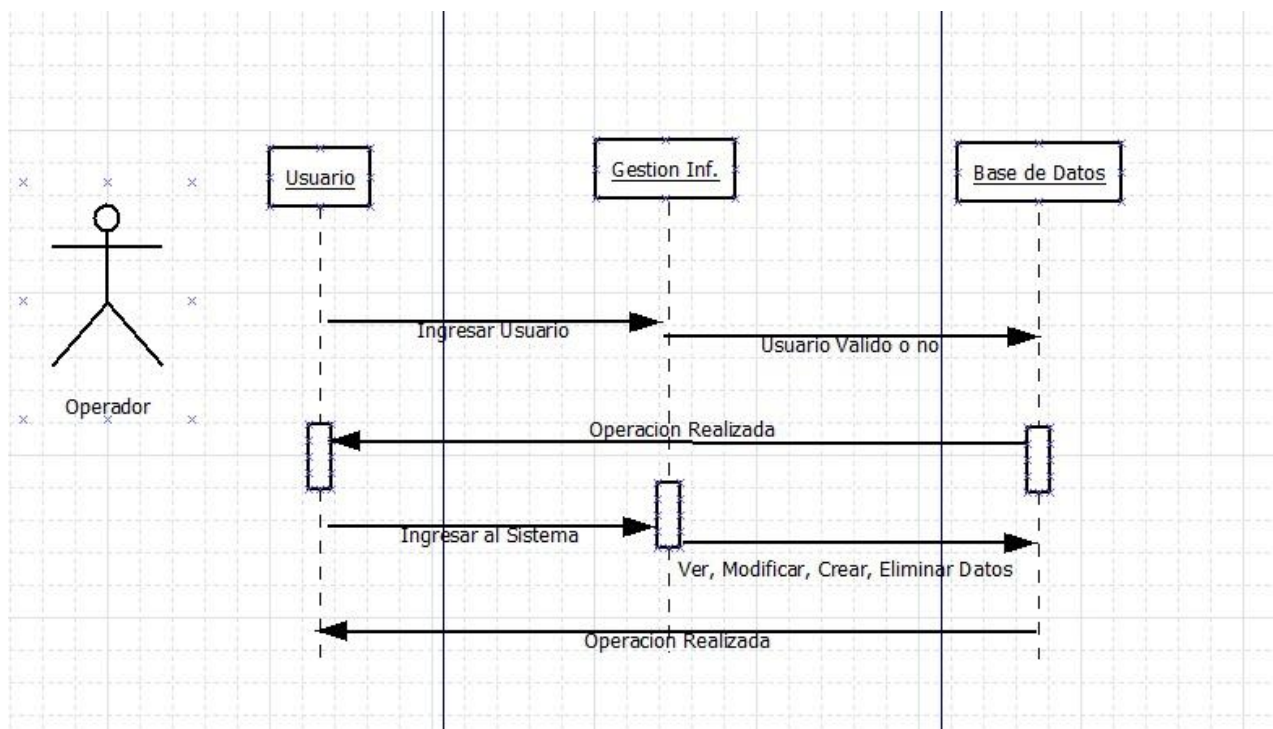
## Diagrama de Clase.

Acá nos encontramos con tres clases las cuales interactúan entre si, para que la clase persona obtenga un rol y su respectivo nombre ya que dentro como fuera de la empresa los diferenciamos como clientes o trabajador, y luego tenemos la clase empresa a la que le asigna al trabajador o cliente un nombre de la empresa a la cual pertenece y un identificador.



## Diagrama de Secuencia.

En este caso, podemos apreciar la interacción del operador con la base de datos mediante el ingreso a la plataforma debido a una validación, si esta validación es correcta, se puede obtener acceso al sistema para realizar las diferentes tareas que este necesite llevar acabo.



**Código en texto plano.**Editar.

```
private void btnActualizar_Click(object sender, EventArgs e)
{
    if (dataGridView1.SelectedRows.Count > 0)
    {
        FrmRegistroCliente FrmC = new FrmRegistroCliente();
        FrmC.txtDni.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
        FrmC.txtApellidos.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
        FrmC.txtNombres.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
        FrmC.txtDireccion.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();
        FrmC.txtTelefono.Text = dataGridView1.CurrentRow.Cells[5].Value.ToString();
        FrmC.txtDni.Focus();
        FrmC.Show();
        if (dataGridView1.SelectedRows.Count > 0)
            Program.Evento = 1;
        else
            Program.Evento = 0;
        dataGridView1.ClearSelection();
    }
    else {
        DevComponents.DotNetBar.MessageBoxEx.Show("Por Favor Seleccione la Fila a Editar.", "Sistema informacion", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    ListarClientes();
}
```



### Buscar.

```
public class clsEmpleado
```

```
{
```

```
    clsManejador M = new clsManejador();
```

```
    public int IdCargo { get; set; }
```

```
    public int IdEmpleado { get; set; }
```

```
    public String Rut{get;set;}
```

```
    public String Apellidos{get;set;}
```

```
    public String Nombres { get; set; }
```

```
    public Char Sexo { get; set; }
```

```
    public DateTime FechaNac { get; set; }
```

```
    public String Direccion { get; set; }
```

```
    public Char EstadoCivil { get; set; }
```

```
    public String MantenimientoEmpleados() {
```

```
        List<clsParametro> lst = new List<clsParametro>();
```

```
        String Mensaje = "";
```

```
        try{
```

```
            lst.Add(new clsParametro("@IdEmpleado", IdEmpleado));
```

```
            lst.Add(new clsParametro("@IdCargo", IdCargo));
```

```
            lst.Add(new clsParametro("@Dni", Rut));
```

```
            lst.Add(new clsParametro("@Apellidos", Apellidos));
```

```
            lst.Add(new clsParametro("@Nombres", Nombres));
```

```
            lst.Add(new clsParametro("@Sexo", Sexo));
```

```
            lst.Add(new clsParametro("@FechaNac", FechaNac));
```

```
            lst.Add(new clsParametro("@Direccion", Direccion));
```

```
            lst.Add(new clsParametro("@EstadoCivil", EstadoCivil));
```

```
        lst.Add(new
clsParametro("@Mensaje","", SqlDbType.VarChar, ParameterDirection.Output, 100));

        M.EjecutarSP("MantenimientoEmpleados", ref lst);

        return Mensaje = lst[9].Valor.ToString();

    }

    catch (Exception ex) {

        throw ex;

    }

}

public DataTable ListadoEmpleados() {

    return M.Listado("ListadoEmpleados", null);

}

public String GenerarIdEmpleado() {

    List<clsParametro> lst = new List<clsParametro>();

    int objIdEmpleado;

    try{

        lst.Add(new
clsParametro("@IdEmpleado","", SqlDbType.Int, ParameterDirection.Output, 4));

        M.EjecutarSP("GenerarIdEmpleado", ref lst);

        objIdEmpleado = Convert.ToInt32(lst[0].Valor.ToString());

    }catch (Exception ex){

throw ex;

    }

    return Convert.ToString(objIdEmpleado);

}

public DataTable BuscarEmpleado(String objDatos)
```

```
{  
    DataTable dt = new DataTable();  
    List<clsParametro> lst = new List<clsParametro>();  
    lst.Add(new clsParametro("@Datos", objDatos));  
    return dt = M.Listado("Buscar_Empleado", lst);  
}  
  
}
```

Crear.

```
public partial class FrmRegistrarEmpleados :
DevComponents.DotNetBar.Metro.MetroForm
{
    clsCargo C = new clsCargo();
    clsEmpleado E = new clsEmpleado();
    int Listado = 0;
    public FrmRegistrarEmpleados()
    {
        InitializeComponent();
    }

    private void FrmRegistrarEmpleados_Load(object sender, EventArgs e)
    {
        timer1.Start();
        timer1.Interval = 1000;
        CargarComboBox();
    }

    private void CargarComboBox(){
        comboBox1.DataSource = C.Listar();
        comboBox1.DisplayMember = "Descripcion";
        comboBox1.ValueMember = "IdCargo";
    }

    private void button1_Click(object sender, EventArgs e)
    {
        FrmRegistrarCargo C = new FrmRegistrarCargo();
        C.Show();
    }
}
```

```
private void btnGrabar_Click(object sender, EventArgs e)
{
    Char EstadoCivil='S';
    E.IdEmpleado = Convert.ToInt32(txtIdE.Text);
    E.IdCargo = Convert.ToInt32(comboBox1.SelectedValue);
    E.Rut = txtDni.Text;
    E.Apellidos = txtApellidos.Text;
    E.Nombres = txtNombres.Text;
    E.Sexo=rbtnMasculino.Checked==true?'M':'F';
    E.FechaNac = Convert.ToDateTime(dateTimePicker1.Value);
    switch (cbxEstadoCivil.SelectedIndex)
    {
        case 1: EstadoCivil = 'S'; break;
        case 2: EstadoCivil = 'C'; break;
        case 3: EstadoCivil = 'D'; break;
        case 4: EstadoCivil = 'V'; break;
    }
    E.EstadoCivil=EstadoCivil;
    E.Direccion = txtDireccion.Text;

    DevComponents.DotNetBar.MessageBoxEx.Show(E.MantenimientoEmpleados(),
"Gestion de Informacion.", MessageBoxButtons.OK, MessageBoxIcon.Information,
MessageBoxDefaultButton.Button1);

    Limpiar();
}

private void Limpiar() {
    cbxEstadoCivil.SelectedIndex = 0;
    txtApellidos.Clear();
    txtDireccion.Clear();
}
```

```
txtDni.Clear();
txtNombres.Clear();
rbtnMasculino.Checked = true;
dateTimePicker1.Value = DateTime.Now;
txtIdE.Clear();
Program.IdCargo = 0;
comboBox1.SelectedIndex = 0;
}

private void timer1_Tick(object sender, EventArgs e)
{
    switch (Listado) {
        case 0: CargarComboBox(); break;
    }
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    timer1.Stop();
}

private void FrmRegistrarEmpleados_Activated(object sender, EventArgs e)
{
    if (Program.IdCargo != 0)
        comboBox1.SelectedValue = Program.IdCargo;
    else
        cbxEstadoCivil.SelectedIndex = 0;
}
}
```

Eliminar.

El sistema presentado como es un prototipo, no hubo implementación de “Eliminar” ya que nos enfocamos principalmente en el almacenamiento y gestión de los datos existentes.