

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN



Programación con estructura de datos

Ing. Santiago José Campos Robles

Grupo 04T

Tema:

Fase 3 - Proyecto de catedra

Alumno	Código
Anthony Eduardo Ortega Cruz	OC200384
Hancel Brandon Alvarado Flores	AF202391
Byron Antonio Méndez Hernández	MH192214
Christian Giovanni Tobar Cerón	TC192221

10 de mayo de 2021

Contenido

Contenido	2
Introducción.....	3
Objetivos	4
Objetivo General.....	4
Objetivos específicos.....	4
Aplicativo de la Estructura de datos seleccionada.....	5
Ambiente de desarrollo	9
Login.....	10
Pruebas de identificación con Roles.....	10
Mostrar módulos.....	13
Desarrollo de mantenimiento y menús	14
Reportes.....	20
Reportes en visual	23
Diagrama Entidad-Relación.....	24
Diccionario de datos.....	24
Registros dentro de entidades.....	28
Tabla Usuarios.....	28
Tabla Solicitudes	28
Tabla Roles.....	28
Tabla Estados.....	29
Tabla Departamento	29
Tabla Casos.....	29
Tabla Bitácora.....	30
Cronograma de actividades.....	30
Conclusión.....	31
Bibliografía	32

Introducción

El motivo por el gran interés dedicado a la gestión y creación de proyectos se hace verídico mediante la evaluación actual de los negocios. Reconociendo que la gestión de proyectos es un sinónimo a una gestión de cambio. Hoy en día las empresas u organizaciones buscan modificar su propio enfoque o dirección de tal modo que implementar cambios veraces requerirá de la incorporación de nuevos procesos, productos y de manera infalible en los costos.

Sin embargo, es preocupante ver un gran número de proyectos que no llegan a satisfacer sus objetivos básicos. Por lo que esto no debe de oscurecer la finalidad relacionada a la creación y gestión de proyectos.

La creación y gestión de proyectos tienen como finalidad fundamental la planificación, el control de actividades, materiales y recursos humanos que participan en un desarrollo de un Sistema de Información. Como resultado es posible identificar los problemas que se producen proporcionando una solución de manera inmediata.

Todo proyecto nace de una necesidad, orientado a la obtención de un resultado dentro de un plazo de tiempo reducido con el fin de poder determinar los recursos y su alcance formado por un conjunto específico de operaciones para alcanzar una meta en común, por ello es habitual que dentro de un equipo de proyecto a menudo se incorporen individuos que no habitualmente no trabajan juntos ya sea por proceder de diferentes áreas.

Esta y otras características hacen que un proyecto deba ser guiado y/o gestionado por especialistas para realizar una entrega a tiempo dentro del presupuesto, un plazo y la calidad con la que se ha planeado además de ofrecer un aporte al aprendizaje. Es donde nace la importancia de una buena gestión de proyectos que recurre a la aplicación de conocimientos, técnicas y habilidades para la realización de un proyecto eficiente y eficaz.

Objetivos

Objetivo General

Crear una solución que ayude a las empresas u organizaciones en la gestión de proyectos de desarrollo, tanto para el inicio de uno nuevo, como para la realización de mejoras de uno existente, por medio del desarrollo de una herramienta de software.

Objetivos específicos

- Definir una serie de pasos para la gestión de proyectos de desarrollo de software con el fin de prevenir una mala planificación y organización a lo largo del seguimiento del requerimiento.
- Mejorar las retroalimentaciones de avance en los proyectos, para evitar inconvenientes por una mala comunicación de las etapas que ya están finalizadas.

Aplicativo de la Estructura de datos seleccionada



Creación de colas que se utilizaran

```
CRUD consultas = new CRUD();  
Validaciones validar = new Validaciones();  
Queue<Departamento> Depa = new Queue<Departamento>();  
Queue<Usuarios> Empleados = new Queue<Usuarios>();
```

Clases enfocadas a la estructura de datos utilizada

```
3 referencias  
class Usuarios  
{  
    1 referencia  
    public string Rol { get; set; }  
    1 referencia  
    public string Departamento { get; set; }  
    1 referencia  
    public string Nombre { get; set; }  
    1 referencia  
    public string Usuario { get; set; }  
    1 referencia  
    public string Contraseña { get; set; }  
}  
  
5 referencias  
class Departamento  
{  
    2 referencias  
    public string nombre { get; set; }  
}
```

Proceso de Encolar la clase destinada

```
if (!validar.validar_Depa(txtNombreDepartamento.Text))
{
    state = false;
    errorProvider1.SetError(txtNombreDepartamento, "Debe de empezar con 'Departamento de' seguido por el área.");
}

if (state)
{
    Departamento depart = new Departamento();
    depart.nombre = txtNombreDepartamento.Text;
    Depa.Enqueue(depart);
    consultas.insertarDepartamento(Depa);
    txtNombreDepartamento.Clear();
    verDepartamento();
    resetear();
}
```

Proceso de desencolamiento e inserción de datos en la BDD SQL Server, los datos pasas como parámetros con sus valores y se agregan junto al commandText para luego ejecutar la consulta y guardarse los datos el gestor de base de datos.

```
}
/* -----DEPARTAMENTO----- */
1 referencia
public void insertarDepartamento(Queue<Departamento> descripcion)
{
    Departamento depa = new Departamento();
    depa = descripcion.Dequeue();
    conexion.conectar();
    SqlCommand comm = new SqlCommand();
    comm.Connection = Conexion.conn;
    comm.CommandText = "Insert into departamento values (@descripcion)";
    comm.Parameters.AddWithValue("descripcion", depa.nombre);
    comm.ExecuteNonQuery();
    conexion.desconectar();
}
```

```
state = false;
errorProvider2.SetError(txtUsername, "Debe ingresar un usuario apropiado. Debe contener solo letras y números.");
}
if (!validar.validar_password(txtContraseña.Text))
{
    state = false;
    errorProvider2.SetError(txtContraseña, "Debe ingresar una contraseña apropiada.");
}

if (state)
{
    Usuarios depart = new Usuarios();
    depart.Rol = cmborol.SelectedValue.ToString();
    depart.Departamento = cmbodepa.SelectedValue.ToString();
    depart.Nombre = txtNombreUsuario.Text;
    depart.Usuario = txtUsername.Text;
    depart.Contraseña = txtContraseña.Text;
    Empleados.Enqueue(depart);
    consultas.insertarUsuario(Convert.ToInt32(cmborol.SelectedValue), Convert.ToInt32(cmbodepa.SelectedValue), txtNombreUsuario);
    verUsuario();
    resetear();
}
```

Uso de arreglos como estructuras estáticas para la generación de códigos al momento de aceptar nueva solicitud de caso.

```
1 referencia
public string GenerarCodigo()
{
    string codigo = "";
    string[] splitDeps;
    splitDeps = departamento.Split(' ');

    codigo = splitDeps[0].Substring(0, 1).ToUpper() + splitDeps[2].Substring(0, 1).ToUpper() + fechasoli.Replace("/", "") + GenerateRandomNo();
    return codigo;
}

1 referencia
public int GenerateRandomNo()
{
    int min = 100;
    int max = 999;
    Random rdm = new Random();
    return rdm.Next(min, max);
}
```

Uso de pilas para guardar datos sobre bitácoras y porcentajes de proyectos.

```
1 referencia
private void actualizar_Click(object sender, EventArgs e)
{
    Stack miPila = new Stack();
    Stack miPila2 = new Stack();
    numericUpDown1.Minimum = numericUpDown1.Value;
    miPila.Push(numericUpDown1);
    miPila2.Push(txtDescripcion.Text);
    consultas.ModificarPorcentajeCaso(numericUpDown1.Value, IDCasos.SelectedValue.ToString());

    SqlDataReader registro = consultas.Registros(IDCasos.SelectedValue.ToString());

    while (registro.Read())
    {
        numericUpDown1.Value = int.Parse(registro["porcentaje_avance"].ToString());
        progressBar1.Value = int.Parse(registro["porcentaje_avance"].ToString());
    }
    conexion.desconectar();
    if (numericUpDown1.Value == 100)
    {
        string x = "";
        int estado = 5;
        x = (IDCasos.SelectedValue.ToString());
        consultas.ModificarEstadoSolicitud(estado, x);
        label10.Show();
        lblEstado.Show();
        lblEstado.Text = "Esperando aprobación de área solicitante";
    }
    else
    {
        label10.Hide();
        lblEstado.Hide();
    }
}
```

Consultas SQL

```
1 referencia
public DataTable verJefesDesarrollo()
{
    conexion.conectar();
    string query = "select id, nombre from usuario where idRol = 3";
    SqlDataAdapter da = new SqlDataAdapter(query, Conexion.conn);
    DataSet ds = new DataSet();
    da.Fill(ds, "Datos");
    conexion.desconectar();
    return ds.Tables["Datos"];
}

1 referencia
public void actualizarSolicitud(int id, string descripcion, DateTime fechaEntrega)
{
    conexion.conectar();
    SqlCommand comm = new SqlCommand();
    comm.Connection = Conexion.conn;
    comm.CommandText = "update solicitud set descripcion = @des, fechaEntrega =@fecha where id = @id";
    comm.Parameters.AddWithValue("id", id);
    comm.Parameters.AddWithValue("des", descripcion);
    comm.Parameters.AddWithValue("fecha", fechaEntrega);
    comm.ExecuteNonQuery();
    conexion.desconectar();
}

//-----PROGRAMADOR-----//

1 referencia
public SqlDataReader Registros(string idcaso)
{
    //leyendo el progreso
    conexion.conectar();
    SqlCommand comando2 = new SqlCommand("select * from caso where id = @id", Conexion.conn);
    comando2.Parameters.AddWithValue("@id", idcaso);
    SqlDataReader registro = comando2.ExecuteReader();
    return registro;
}

1 referencia
public void eliminarUsuario(int id)
{
    conexion.conectar();
    SqlCommand comm = new SqlCommand();
    comm.Connection = Conexion.conn;
    comm.CommandText = "delete from usuario where id=@id";
    comm.Parameters.AddWithValue("id", id);
    comm.ExecuteNonQuery();
    conexion.desconectar();
}

//-----OTROS-----//

2 referencias
public DataTable verCasos(int idUsuario)
{
    conexion.conectar();
    SqlCommand cmd = new SqlCommand("select id,descripcion from caso where programador=" + idUsuario, Conexion.conn);
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    DataTable dt = new DataTable();
    da.Fill(dt);
}
```


Ambiente de desarrollo

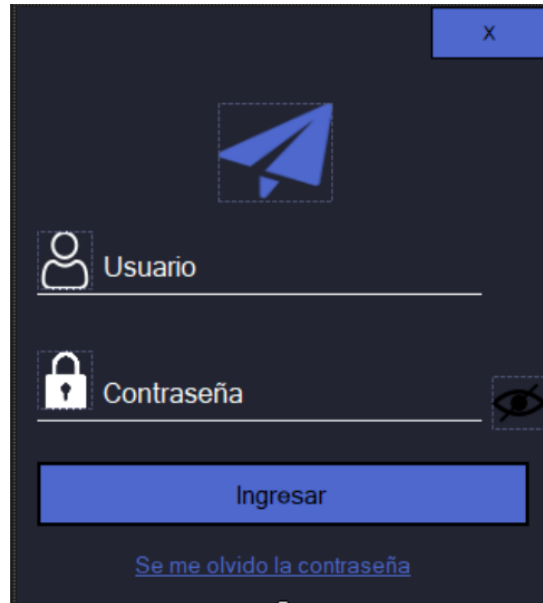
Como grupo hemos visto a bien utilizar el ambiente de formularios para realizar a cabo el proyecto, principalmente por las facilidades de personalización y la amplia variedad de herramientas que ofrece esta interfaz para el desarrollo de tareas específicas que se desarrollan dentro del proyecto.

A esto le sumamos que al trabajar de esta manera nos percatamos como se desenvuelve el proyecto de cara a los usuarios de este, ya que el ambiente de formularios da un ambiente familiar y amigable a los usuarios de aplicaciones de escritorio.

Nosotros como grupo hemos llegado a decidir que el ambiente más óptimo para el desarrollo de nuestro proyecto es en **Ambiente de escritorio “más concretamente en una Ambiente de Formularios”** ya que nos brinda las herramientas y oportunidades de dar a lucir nuestro proyecto de **CASOLITYCAL** simulando la implementación de dicho proyecto ya en un entorno laboral sin dejar de lado que previamente analizamos los pros y contras que trae el desarrollarlo en un ambiente como el que hemos escogido y a su vez sumarle el objetivo que tenemos y es el implementar las estructuras de datos dándoles un buen e importante uso dentro de la aplicación lo cual en un futuro nos facilite el mantenimiento y control de datos dentro del sistema, también todos los miembros participantes llegamos a ciertas conclusiones al optar el desarrollo en un ambiente de escritorio y ocupar un ambiente de formularios ya que una app desarrollada aquí brinda mayor robustez y un mejor tiempo de respuesta así también abre un mar de posibilidades futuras al pensar que nos da la facultad de realizar cualquier cambio a cosas siempre y cuando el software nos lo permita (cuestión gráfica, control total de las entradas del usuario al momento de capturar) de igual modo no podemos dejar de lado que entre sus desventajas más notorias esta que la portabilidad de la aplicación no es la mejor ya que no es conveniente si nuestra empresa contara con muchos usuarios los cuales estarán accediendo desde diferentes dispositivos pero a pesar de todo eso planeamos como empresa y futuros programadores desarrollar –dar un buen uso e implementación de las estructuras de datos dándole a si al usuario final la mejor experiencia al visitar nuestro proyecto

Login.

El inicio de sesión se realizará con los datos registrados por el usuario administrador en el se ingresará en nombre de usuario y la contraseña, si el usuario desea ver la contraseña podrá presionar sobre el botón del lado derecho.

A login form with a dark blue background. At the top right is a close button with an 'X'. Below it is a logo of a blue paper airplane. The form has two input fields: 'Usuario' with a person icon and 'Contraseña' with a lock icon. To the right of the password field is an eye icon for toggling visibility. Below the fields is a large blue 'Ingresar' button. At the bottom is a link that says 'Se me olvido la contraseña'.

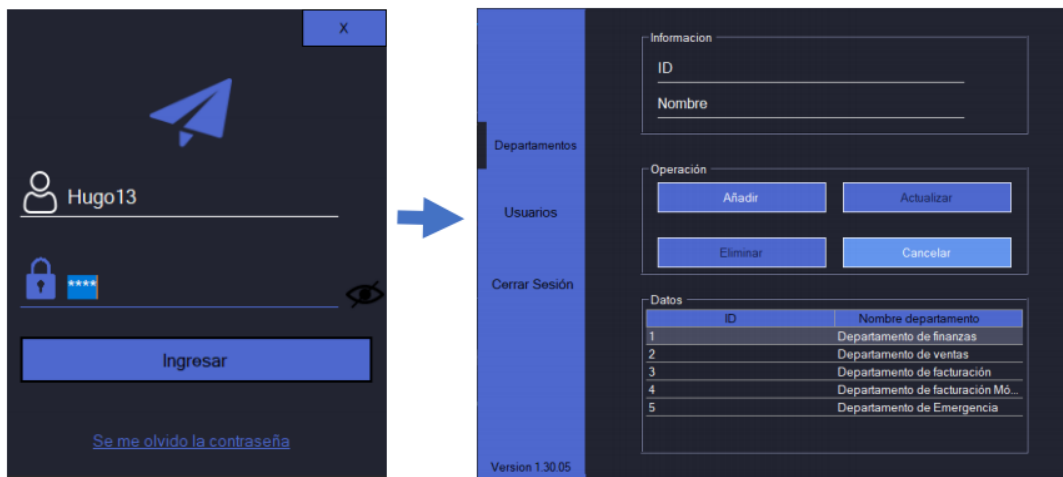
Pruebas de identificación con Roles.

Con la siguiente tabla de usuario se accederá y así se dará privilegios de acuerdo a su idRol.

	id	descripcion
▶	1	Administrador
	2	Jefe de áreas funcionales
	3	Jefe de desarrollo
	4	Empleado de las áreas funcionales
	5	Programador
*	NULL	NULL

	id	idRol	idDepartamento	nombre	usuario	pass
▶	1	1	4	Hugo Vides	Hugo13	1234
	2	2	4	Pablo Acosta	Pablo14	14Pablo
	3	3	4	Nicole Cruz	Nicole14	1234
	4	4	4	Erick	Erick14	1234
	5	5	4	Diego Martinez	Dieg14	1234
*	NULL	NULL	NULL	NULL	NULL	NULL

Accediendo como Administrador

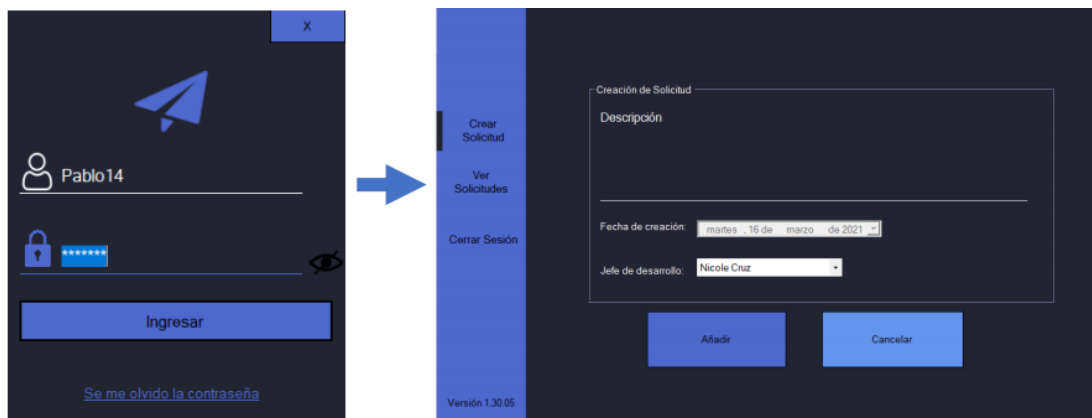


The image shows a login screen on the left and an administrator dashboard on the right, connected by a blue arrow. The login screen has a Telegram logo, a username field with 'Hugo13', a password field with a lock icon and a 'Ver contraseña' (Show password) eye icon, an 'Ingresar' (Login) button, and a link 'Se me olvidó la contraseña' (I forgot my password). The dashboard on the right has a sidebar with 'Departamentos', 'Usuarios', and 'Cerrar Sesión' (Logout). The main area contains a form for adding or updating users with fields for 'ID' and 'Nombre', buttons for 'Añadir', 'Actualizar', 'Eliminar', and 'Cancelar', and a table of departments.

ID	Nombre departamento
1	Departamento de finanzas
2	Departamento de ventas
3	Departamento de facturación
4	Departamento de facturación Mó.
5	Departamento de Emergencia

Version 1.30.05

Accediendo como jefe de áreas funcionales



The image shows a login screen on the left and a functional area manager dashboard on the right, connected by a blue arrow. The login screen has a Telegram logo, a username field with 'Pablo14', a password field with a lock icon and a 'Ver contraseña' (Show password) eye icon, an 'Ingresar' (Login) button, and a link 'Se me olvidó la contraseña' (I forgot my password). The dashboard on the right has a sidebar with 'Crear Solicitud', 'Ver Solicitudes', and 'Cerrar Sesión' (Logout). The main area contains a form for creating a request with a 'Descripción' field, a 'Fecha de creación' (Creation date) field with a date picker set to 'martes 16 de marzo de 2021', a 'Jefe de desarrollo' (Development manager) dropdown menu with 'Nicole Cruz' selected, and buttons for 'Añadir' (Add) and 'Cancelar' (Cancel).

Version 1.30.05

```

private void button1_Click(object sender, EventArgs e)
{
    errorProvider1.Clear(); //Limpiar ErrorProvider
    bool state = true;

    if (!validar.validar_usuario(textBox1.Text))
    {
        state = false;
        errorProvider1.SetError(textBox1, "Debe ingresar un usuario apropiado. Ejemplo: Eduard13");
    }
    if (!validar.validar_password(textBox2.Text))
    {
        state = false;
        errorProvider1.SetError(textBox2, "Debe ingresar una contraseña apropiada.");
    }

    if (state)
    {
        string tipo = consultas.autenticarse(textBox1.Text, textBox2.Text);
        int idLogeado = consultas.idUsuarioLogeado(textBox1.Text, textBox2.Text);
        switch (tipo)
        {
            case "1":
                this.Dispose();
                Administrador admin = new Administrador();
                admin.ShowDialog();
                break;
            case "2":
                this.Dispose();
                JefeAreaFuncional JAF = new JefeAreaFuncional(idLogeado);
                JAF.ShowDialog();
                break;
            case "3":
                this.Dispose();
                JefeDesarrollo JD = new JefeDesarrollo(idLogeado);
                JD.ShowDialog();
                break;
            case "4":
                this.Dispose();
                tester frmTester = new tester(idLogeado);
                frmTester.ShowDialog();
                break;
            case "5":
                this.Dispose();
                Programador coder = new Programador(idLogeado);
                coder.ShowDialog();
                break;
            default:
                label1.Text = "Usuario no válido";
                break;
        }
    }
}

```

Mostrar módulos

- **Modulo administrador**
Con la capacidad de registrar y gestionar áreas funcionales de la empresa (departamentos), jefes de áreas funcionales y jefes de desarrollo.
- **Modulo jefes de áreas funcionales**
Con la capacidad de solicitar la apertura de casos y monitorear el porcentaje de progreso y bitácora de los casos aperturados.
- **Modulo empleados de las áreas funcionales**
Pueden ser asignados como “probadores” de un caso, en tal situación deben monitorear el porcentaje de progreso y bitácoras de trabajo de dicho caso y aprobarlo o rechazarlo una vez que este haya sido entregado por el programador asignado.
- **Modulo jefe de desarrollo**
Tiene la capacidad de aceptar o rechazar las solicitudes de casos realizadas por los jefes de las áreas funcionales que tiene a su cargo. Además, debe tener la capacidad de monitorear el trabajo de los programadores que tiene a su cargo.
- **Modulo programadores**
Debe actualizar los porcentajes de progreso y bitácoras de trabajo de los casos a los que ha sido asignado.

Desarrollo de mantenimiento y menús

Departamentos

El usuario tipo “Administrador” tendrá acceso a la creación de departamentos, ejemplo “Departamento de marketing” dentro de cada departamento tendrá su tipo de área de trabajo y administración de proyectos. El administrador registra el nombre del departamento y el sistema genera un código o ID auto incrementable.

The screenshot shows the 'Departamentos' management interface. On the left is a sidebar with 'Departamentos', 'Usuarios', and 'Cerrar Sesión'. The main area has three sections: 'Información' with fields for 'ID' and 'Nombre'; 'Operación' with buttons for 'Añadir', 'Actualizar', 'Eliminar', and 'Cancelar'; and 'Datos' with a table of existing departments. The 'Añadir' button is highlighted, indicating the 'Agregar' operation.

id	descripcion
1	Departamento de finanzas
2	Departamento de ventas
3	Departamento de facturación
4	Departamento de facturación Móvil

Version 1.30.03

This screenshot shows the 'Actualizar' operation. The 'ID' field is populated with '1' and the 'Nombre' field with 'Departamento de marketing'. The 'Actualizar' button is highlighted. The 'Datos' table remains the same as in the previous screenshot.

id	descripcion
1	Departamento de finanzas
2	Departamento de ventas
3	Departamento de facturación
4	Departamento de facturación Móvil

Version 1.30.03

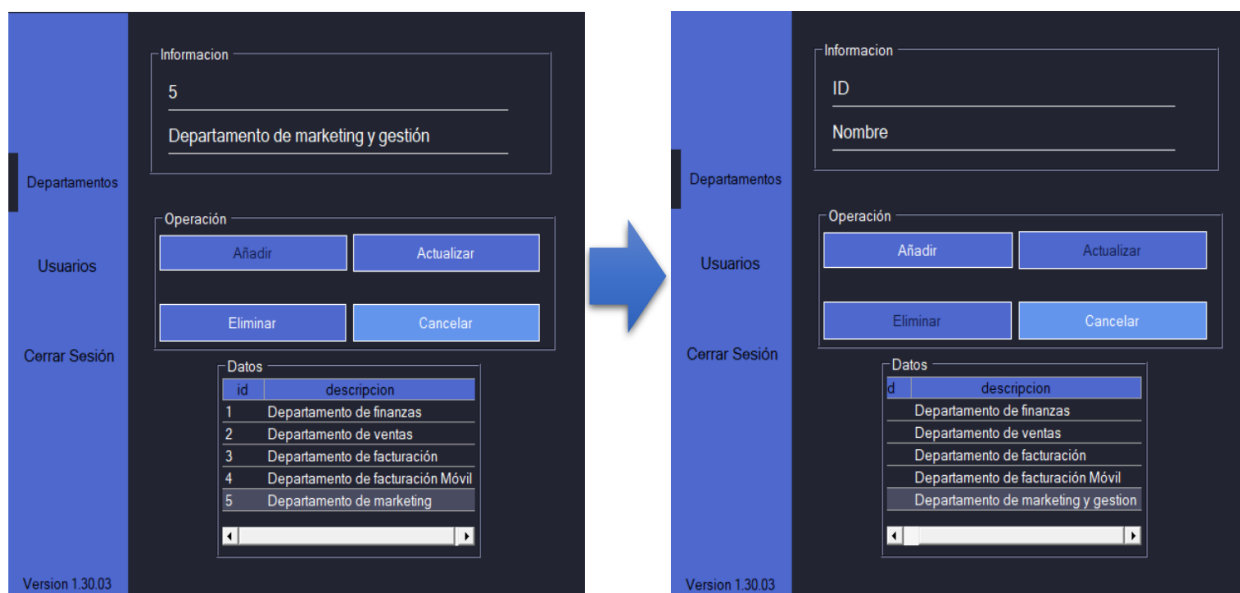


This screenshot shows the 'Agregar' operation. The 'ID' field is empty and the 'Nombre' field contains 'Departamento de marketing'. The 'Agregar' button is highlighted. The 'Datos' table now includes the new department as a fifth row.

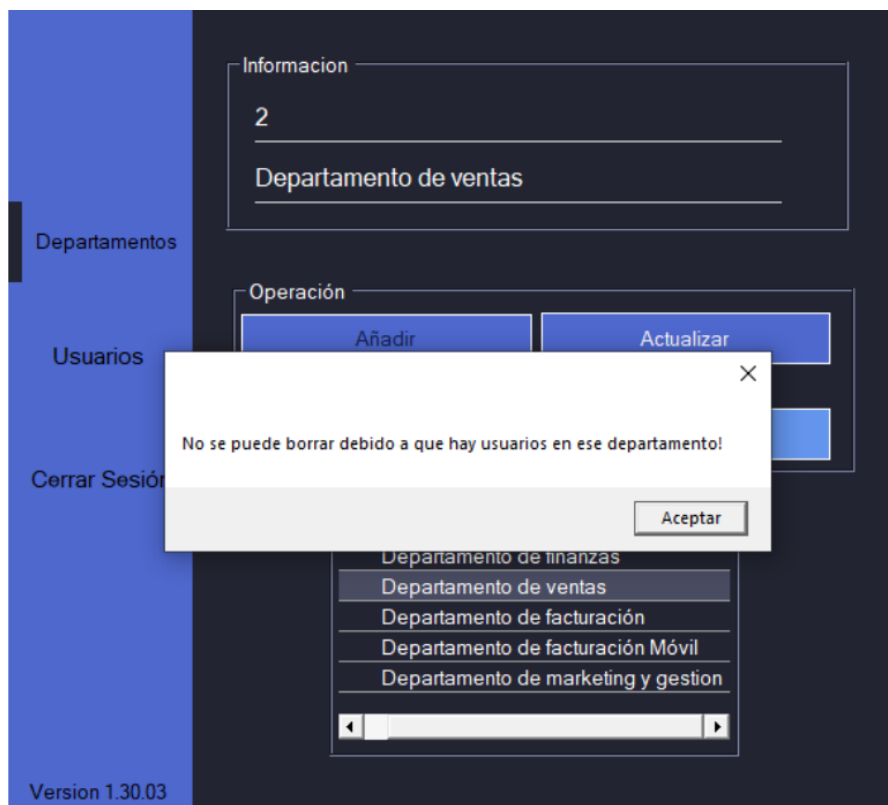
id	descripcion
1	Departamento de finanzas
2	Departamento de ventas
3	Departamento de facturación
4	Departamento de facturación Móvil
5	Departamento de marketing

Version 1.30.03

Para editar los departamentos que se encuentran dentro del datagridview el usuario “Administrador” deberá dar doble click en la fila que desea modificar. Luego realizar los cambios según el nombre y presionar el botón “Actualizar”.



Finalmente, si el usuario “Administrador” decide eliminar cualquier departamento este podrá siempre y cuando no existan otro tipo de usuario dentro de él, caso contrario mostrará un mensaje de error.



Para eliminar un departamento se requiere de la misma función realizada al momento de editar un departamento, en este caso deberá dar doble click en la fila o departamento a eliminar y presionar el botón “Eliminar” para ejercer los cambios.

Información

ID

Nombre

Operación

Añadir Actualizar

Eliminar Cancelar

Datos

id	descripcion
1	Departamento de finanzas
2	Departamento de ventas
3	Departamento de facturación
4	Departamento de facturación Móvil

Version 1.30.03

Para cancelar cambios al momento que el usuario “Administrador” haya seleccionado solo deberá presionar en el botón “Cancelar” para limpiar datos escritos en los textbox

Usuarios

El usuario “Administrador” podrá añadir más usuarios con los siguientes datos:

- Tipo de usuario.
- Departamento (explicado en el punto anterior).
- Nombre de la persona.
- Nombre de usuario o Username.
- Contraseña.

Información					
ID					
Jefe de áreas funcionales					
Departamento de finanzas					
Nombre					
Usuario					
Contraseña					

Operación					
Añadir					
Actualizar					
Eliminar					
Cancelar					

Datos					
id	idRol	idDepartamento	nombre	usuario	
1	1	2	Hugo Vides	Admin12	
2	2	2	Marcelo Benitez	Marce12	
3	3	2	Antony Perez	Anton12	
4	4	2	Ricardo Rivera	River12	
5	5	2	Bryan Lopez	Lopez12	

Version 1.30.03

El usuario “Administrador” podrá agregar más usuarios administradores, pero no podrá modificarlos, mostrará un mensaje de advertencia. Para añadir más usuarios el usuario encargado ingresará los datos mencionados con anterioridad y presionará en el botón añadir, los botones de actualizar y eliminar se encuentran bloqueados por el hecho que no se encuentran usuarios seleccionados. Cada campo contará con ciertas validaciones que deberán cumplirse.

- El nombre solo debe incluir caracteres de letras, sin incluir números ni símbolos.
- El usuario deberá incluir letras y números un mínimo de 6 caracteres y un máximo de 15 caracteres a escribir.
- La contraseña debe incluir de 4 a 8 caracteres alfanuméricos.

Departamentos

Usuarios

Cerrar Sesión

Información

ID

Jefe de desarrollo

Departamento de facturación

Alejandra Perez

AleP12

1234

Operación

Añadir

Actualizar

Eliminar

Cancelar

Datos

id	idRol	idDepartamento	nombre	usuario
1	1	2	Hugo Vides	Admin12
2	2	2	Marcelo Benitez	Marce12
3	3	2	Antony Perez	Anton12
4	4	2	Ricardo Rivera	River12
5	5	2	Bryan Lopez	Lopez12

Version 1.30.03

Departamentos

Usuarios

Cerrar Sesión

Información

ID

Jefe de áreas funcionales

Departamento de finanzas

Nombre

Usuario

Contraseña

Operación

Añadir

Actualizar

Eliminar

Cancelar

Datos

id	idRol	idDepartamento	nombre	usuario
1	1	2	Hugo Vides	Admin12
2	2	2	Marcelo Benitez	Marce12
3	3	2	Antony Perez	Anton12
4	4	2	Ricardo Rivera	River12
5	5	2	Bryan Lopez	Lopez12
6	3	4	Alejandra Perez	AleP12

Version 1.30 03

Para editar los usuarios que se encuentran dentro del datagridview, el usuario “Administrador” deberá dar doble click en la fila que desea modificar. Luego realizar los cambios según el nombre de persona, nombre de usuario o username y contraseña para después presionar el botón “Actualizar” de igual forma contará con las validaciones mencionadas con anterioridad.

Departamentos

Usuarios

Cerrar Sesión

Información

6

Alejandra Perez

PerezAle100

AP1234

Operación

Añadir

Actualizar

Eliminar

Cancelar

Datos

id	idRol	idDepartamento	nombre	usuario
1	1	2	Hugo Vides	Admin12
2	2	2	Marcelo Benitez	Marce12
3	3	2	Antony Perez	Anton12
4	4	2	Ricardo Rivera	River12
5	5	2	Bryan Lopez	Lopez12
6	3	4	Alejandra Perez	AleP12

Version 1.30.03

Información		Operación	
ID		Añadir	
Jefe de áreas funcionales	-	Actualizar	
Departamento de finanzas	-	Eliminar	
Nombre		Cancelar	
Usuario			
Contraseña			

Datos				
id	idDepartamento	nombre	usuario	pass
2		Hugo Vides	Admin12	1234
2		Marcelo Benitez	Marce12	1234
2		Antony Perez	Anton12	1234
2		Ricardo Rivera	River12	1234
2		Bryan Lopez	Lopez12	1234
4		Alejandra Perez	PerezAle100	AP1234

Version 1.30.03

Para eliminar un usuario se requiere de la misma función realizada al momento de editar un usuario, en este caso deberá dar doble click en la fila o usuario a eliminar y presionar el botón “Eliminar” para ejercer los cambios.

The diagram illustrates the process of deleting a user. It shows two states of the application interface, connected by a blue arrow indicating a transition.

Left Interface (Agregar - Add):

- Operación:** Añadir
- Información:**
 - ID: 6
 - Jefe de desarrollo
 - Departamento de facturad
 - Nombre: Alejandra Perez
 - Usuario: PerezAle100
 - Contraseña: AP1234
- Datos (Table):**

id	idRol	idDepartamento	nombre	usuario
1	1	2	Hugo Vides	Admin12
2	2	2	Marcelo Benitez	Marce12
3	3	2	Antony Perez	Anton12
4	4	2	Ricardo Rivera	River12
5	5	2	Bryan Lopez	Lopez12
6	3	4	Alejandra Perez	PerezAle1

Right Interface (Eliminar - Delete):

- Operación:** Eliminar
- Información:**
 - ID
 - Jefe de áreas funcionales
 - Departamento de finanzas
 - Nombre
 - Usuario
 - Contraseña
- Datos (Table):**

id	idRol	idDepartamento	nombre	usuario
1	1	2	Hugo Vides	Admin12
2	2	2	Marcelo Benitez	Marce12
3	3	2	Antony Perez	Anton12
4	4	2	Ricardo Rivera	River12
5	5	2	Bryan Lopez	Lopez12

Para cancelar cambios al momento que el usuario “Administrador” haya seleccionado solo deberá presionar en el botón “Cancelar” para limpiar datos escritos en los textbox. Para concluir con el apartado de administrador este tipo de usuario podrá Cerrar sesión a lo cual lo enviará al formulario de Inicio de sesión.

Reportes



Reporte de Bitácoras (Casolitycal)

id	id Caso	Descripcion
1	DF30102020851	Añadido login
2	DF30102020851	Añadido web service
3	DF852021846	Creación del login
4	DF852021846	Proyecto terminado
5	DF852021846	Control de usuarios incorporado
1003	DF852021650	Creación de web services
1004	DF852021650	Creación de control de departamentos
1005	DF852021650	Identificación de usuarios mediante Login
1006	DF852021821	Modificación de registro de usuarios
1007	DF852021821	Creación de modelo de BDD
1008	DF852021821	Mejora de servidores locales
1009	DF852021821	Identificación de recursos utilizados



Reporte de Usuarios (Casolitycal)

id	id Rol	Id Departamento	Nombre	Usuario	Contraseña
1	1	4	Hugo Vides	Hugo13	1234
2	2	4	Pablo Acosta	Pablo14	14Pablo
3	3	4	Nicole Cruz	Nicole14	1234
4	4	4	Erick	Erick14	1234
5	5	4	Diego Martinez	Dieg14	1234
6	5	5	David Perez	David02	1234



Reporte de Casos (Casolitycal)

id	id Solicitud	descripción	Porcentaje avance	Observaciones	Fecha limite	Tester	Programador	Fecha produccion
DF30102020851	1	Uso de JS	100	Finalizado	06/11/2020 17:25:30	4	5	06/11/2020 17:31:27
DF852021453	1006	Uso de Ruby on Rails acompañado con MYSQL Lite	14		27/05/2021 18:21:56	4	5	
DF852021581	1008	Obtener record crediticio y cálculos de intereses anuales	0		27/05/2021 18:21:56	4	6	
DF852021650	1005	Uso de HTML, CSS y PHP validado con funciones REGEX en JS	100		27/05/2021 18:21:56	4	5	
DF852021821	1003	Uso de C# en entorno web(ASP.Net)	3		17/05/2021 18:21:56	4	5	
DF852021846	3	-Trabaje en SQL SERVER -JS, PHP y HTML	100	Finalizado	15/05/2021 15:16:28	4	6	16/06/2021 15:17:29
DF852021918	1004	Uso de JS	83		22/05/2021 18:21:56	4	5	

Reportes en visual

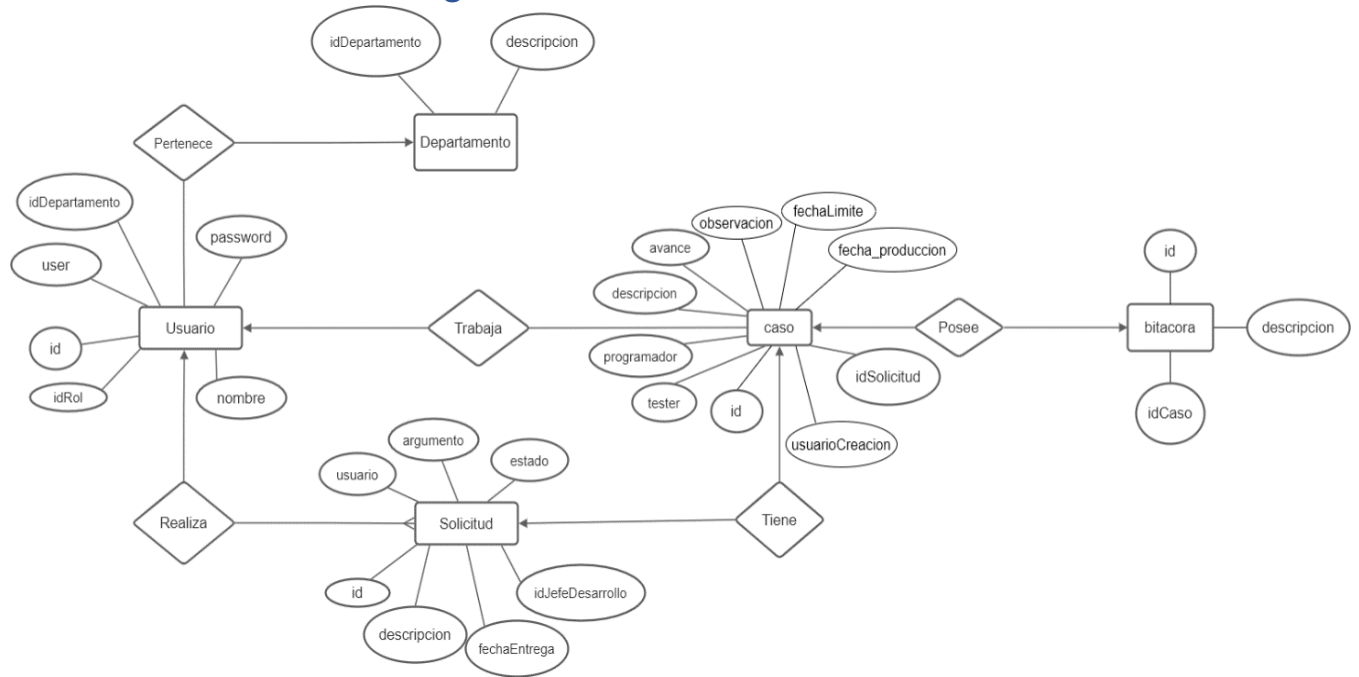
The screenshot shows the Visual Studio IDE with the 'Reporte de Casos (Casolitycal)' report open. The report is a table with 9 columns: id, id Solicitud, Descripción, Porcentaje avance, Observaciones, Fecha limite, Tester, Programador, and Fecha producción. The table contains 5 rows of data. The left sidebar shows the 'Datos de informe' section with 'Reporte Bitacora.rdl [Diseño]', 'Reporte Caso.rdl [Diseño]', and 'Reporte Usuario.rdl [Diseño]'. The right sidebar shows the 'Explorador de soluciones' section with 'Solución "Reportes PED" (1 de 1)' and 'Reportes PED' folder containing 'Orígenes de datos comp', 'Conjuntos de datos comp', 'Informes', 'Reporte Bitacora.rdl', 'Reporte Caso.rdl', and 'Reporte Usuario.rdl'. The bottom status bar shows 'Listo' and 'Agregar al control de código fuente'.

id	id Solicitud	Descripción	Porcentaje avance	Observaciones	Fecha limite	Tester	Programador	Fecha producción
DF3010202085	1	Uso de JS	100	Finalizado	06/11/2020 17:25:30	4	5	06/11/2020 17:31:27
DF852021453	1006	Uso de Ruby on Rails acompañado con MySQL Lite	14		27/05/2021 18:21:56	4	5	
DF852021581	1008	Obtener record crediticio y cálculos de intereses anuales	0		27/05/2021 18:21:56	4	6	
DF852021650	1005	Uso de HTML, CSS y PHP validado con funciones REGEX en JS	100		27/05/2021 18:21:56	4	5	
DF852021821	1003	Uso de C# en entorno web (ASP .Net)	3		17/05/2021 18:21:56	4	5	

The screenshot shows the Visual Studio IDE with the 'Reporte de Bitacoras (Casolitycal)' report open. The report is a table with 3 columns: id, id Caso, and Descripción. The table contains 10 rows of data. The left sidebar shows the 'Datos de informe' section with 'Reporte Bitacora.rdl [Diseño]', 'Reporte Caso.rdl [Diseño]', and 'Reporte Usuario.rdl [Diseño]'. The right sidebar shows the 'Explorador de soluciones' section with 'Solución "Reportes PED" (1 de 1)' and 'Reportes PED' folder containing 'Orígenes de datos comp', 'Conjuntos de datos comp', 'Informes', 'Reporte Bitacora.rdl', 'Reporte Caso.rdl', and 'Reporte Usuario.rdl'. The bottom status bar shows 'Listo' and 'Agregar al control de código fuente'.

id	id Caso	Descripción
1	DF3010202085	Añadido login
2	DF3010202085	Añadido web service
3	DF 852021840	Creación del login
4	DF 852021840	Proyecto terminado
5	DF 852021840	Control de usuarios incorporado
1003	DF 852021850	Creación de web service
1004	DF 852021850	Creación de control de departamentos
1006	DF 852021860	Identificación de usuarios mediante Login
1006	DF 852021821	Modificación de registro de usuarios
1007	DF 852021821	Creación de modelo de BDD
1008	DF 852021821	Mejora de servidores locales
1009	DF 852021821	Identificación de recursos utilizados

Diagrama Entidad-Relación



Diccionario de datos

El diseño de la base de datos estará representado por 7 tablas las cuales cada una tendrá una función diferente:

- Caso.
- Solicitud.
- Bitácora
- Departamento
- Usuario.
- Rol
- Estados

El nombre de las clases les fue colocado teniendo en cuenta qué es lo que cada una de ellas realizará dentro del proyecto, para detallar un poco más estas empezamos con la clase Usuario.

- **Tabla rol:** en ellas representamos los diferentes roles que tendrá cada uno de los usuarios:
 1. Id: identificador único de cada rol que se tenga dentro del proyecto.
 2. Descripción: nombre del campo otorgado ejemplo: Administrador, programador, etc

Tabla Estado: contendrá los diferentes avisos o estados que dispondrá en solicitudes, casos, bitácoras, etc.

1. Id_estado: identificador que tendrá la tabla estado
2. Nombre: descripción del caso. Ejemplo: Vencido, en espera, etc.

- **Tabla Usuario:** en ella representamos a los diferentes tipos de usuarios que tienen acceso al sistema, se guardan principalmente datos generales de una persona los que definimos en sus atributos:

1. Id: identificador único de cada usuario que se registre dentro del proyecto.
2. idDepartamento: este nos permite conocer de forma fácil y rápida saber a qué departamento pertenece ese usuario.
3. idRol: identificador encargado de referirnos a los tipos de roles dentro del programa y describir qué tipo de usuario es la persona registrada.
4. Nombre: nombre del empleado registrado sin importar su rol.
5. Usuario: es el medio con el cuál se identifican antes de entrar al programa.
6. Password: contraseña registrada que le da acceso a cada usuario cuando quieran hacer uso del sistema.

Esta clase no cuenta con métodos, ya que solo describe las generalidades de una persona, en las especializaciones de esta clase son las que detallan qué funciones tiene dentro de la plataforma.

- **Tabla Departamento:** Clase encargada de representar a los departamentos que están dentro de la empresa, no realizan ninguna función dentro del programa sus atributos son:

1. idDepartamento: Identificador único de cada uno de los departamentos.
2. Descripción: contiene el nombre del departamento.

- **Tabla Solicitud:** clase que almacena y describe las solicitudes hechas.

1. Id: identificador único asociado a cada una de las solicitudes creadas.

2. Descripción: contiene el detalle de lo que se ha solicitado, describe la problemática y las necesidades necesitan ser cubiertas.
 3. fechaEntrega: es la fecha en que se crea la solicitud.
 4. idJefeDesarrollo: identificador del jefe de desarrollo al que se le asignó la solicitud.
 5. Estado: la disponibilidad de la solicitud, describe si esta está en espera, rechazada o aprobada.
 6. usuarioCreacion: el id del jefe de área funcional que solicitó el desarrollo de la solución.
 7. Argumento: en caso de ser rechazada la solicitud se debe de poner una razón por la cual se ha rechazado la solicitud.
- **Tabla Caso:** Esta clase guarda el detalle del avance de las soluciones que estén en desarrollo.
 1. Id: identificador único de cada uno de los casos con un formato específico (3 letras, la fecha y tres números), la fecha en que se realizó la solicitud formato (YY) y un número de tres cifras generado aleatoriamente.
 2. idSolicitud: es el identificador de la solicitud de la cuál proviene el desarrollo de la solución.
 3. Descripción: descripción de la solicitud detallada con los comentarios del jefe de desarrollo.
 4. porcentajeAvance: campo que registra el avance de desarrollo de las soluciones.
 5. fechaLimite: Viene dado por el jefe de desarrollo, es la fecha en la que la solución debe de ser entregada al tester designado.
 6. usuarioCreacion: almacena el identificador del usuario que crea el caso.
 7. Tester: el campo almacena la referencia del usuario designado como tester una vez esté terminada la solución solicitada.
 8. Programador: se refiere al id del programador que el jefe de desarrollo puso como encargado del proyecto.

9. Fecha_produccion: se registra a la fecha en la que se acepta la solución y se pone a disposición del departamento.
- **Tabla bitácora:** Contiene los cambios realizados por los programadores, sobre el avance de los proyectos.
 1. Id: identificador de las bitácoras asociado a cada una de ellas.
 2. idCaso: se refiere al id del caso al cual pertenece la bitácora.
 3. Descripción: es el campo donde se actualiza periódicamente el avance del proyecto.

Relaciones:

Entre la clase usuario y sus hijas existe herencias, desde la clase padre Usuario hacia las demás clases derivadas o las especializaciones de esta misma.

La relación que existe entre Departamento y Usuario es de muchos a uno, donde muchos usuarios pertenecen a un departamento y un departamento puede tener muchos usuarios.

La Clase caso se relaciona con la solicitud con una cardinalidad de uno a uno, lo que quiere decir que una solicitud tiene un único caso y un caso posee una única solicitud.

Entre la clase caso y bitácora hay una relación de composición, ya que la una bitácora no puede existir sin un caso

Registros dentro de entidades

Tabla Usuarios

LAPTOP-1BPNONU4\...OO - dbo.usuario						
	id	idRol	idDepartamento	nombre	usuario	pass
▶	1	1	4	Hugo Vides	Hugo13	1234
	2	2	4	Pablo Acosta	Pablo14	14Pablo
	3	3	4	Nicole Cruz	Nicole14	1234
	4	4	4	Erick	Erick14	1234
	5	5	4	Diego Martinez	Dieg14	1234
	6	5	5	David Perez	David02	1234
*	NULL	NULL	NULL	NULL	NULL	NULL

Tabla Solicitudes

LAPTOP-1BPNONU4\...O - dbo.solicitud							
LAPTOP-1BPNONU4\...OO - dbo.usuario							
	id	descripcion	fechaEntrega	idJefeDesarrollo	estado	usuarioCreacion	argumento
▶	1	Creación de pr...	2020-10-30 17:1...	3	6	2	En Desarrollo
	2	Tienda Online d...	2020-10-30 17:1...	3	3	2	denegado por f...
	3	Creación de un ...	2021-05-08 15:0...	3	8	2	En Desarrollo
	1003	Sistema de Res...	2021-05-08 18:2...	3	2	2	En Desarrollo
	1004	Programa para ...	2021-05-08 18:2...	3	2	2	En Desarrollo
	1005	Creación de pr...	2021-05-08 18:2...	3	5	2	En Desarrollo
	1006	Programa de E...	2021-05-08 18:2...	3	2	2	En Desarrollo
	1007	Programa de G...	2021-05-08 18:2...	3	1	2	Esperando Revi...
	1008	Aplicación mov...	2021-05-08 18:2...	3	2	2	En Desarrollo
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tabla Roles

LAPTOP-1BPNONU4\...to_POO - dbo.rol		
LAPTOP-1BPNONU4\...O - dbo.solicitud		
LAPTOP-1BPNONU4\...OO - dbo.usuario		
	id	descripcion
▶	1	Administrador
	2	Jefe de áreas funcionales
	3	Jefe de desarrollo
	4	Empleado de las áreas funcion...
	5	Programador
*	NULL	NULL

Tabla Estados

	id_estado	nombre
▶	1	En espera de respuesta
	2	Solicitud aceptada
	3	Solicitud rechazada
	4	En desarrollo
	5	Esperando aprobación de área solicitante
	6	Vencido
	7	Devuelto con observaciones
	8	Finalizado
*	NULL	NULL

Tabla Departamento

	id	descripcion
▶	1	Departamento de finanzas
	2	Departamento de ventas
	3	Departamento de facturación
	4	Departamento de facturación Móvil
	5	Departamento de Emergencia
*	NULL	NULL

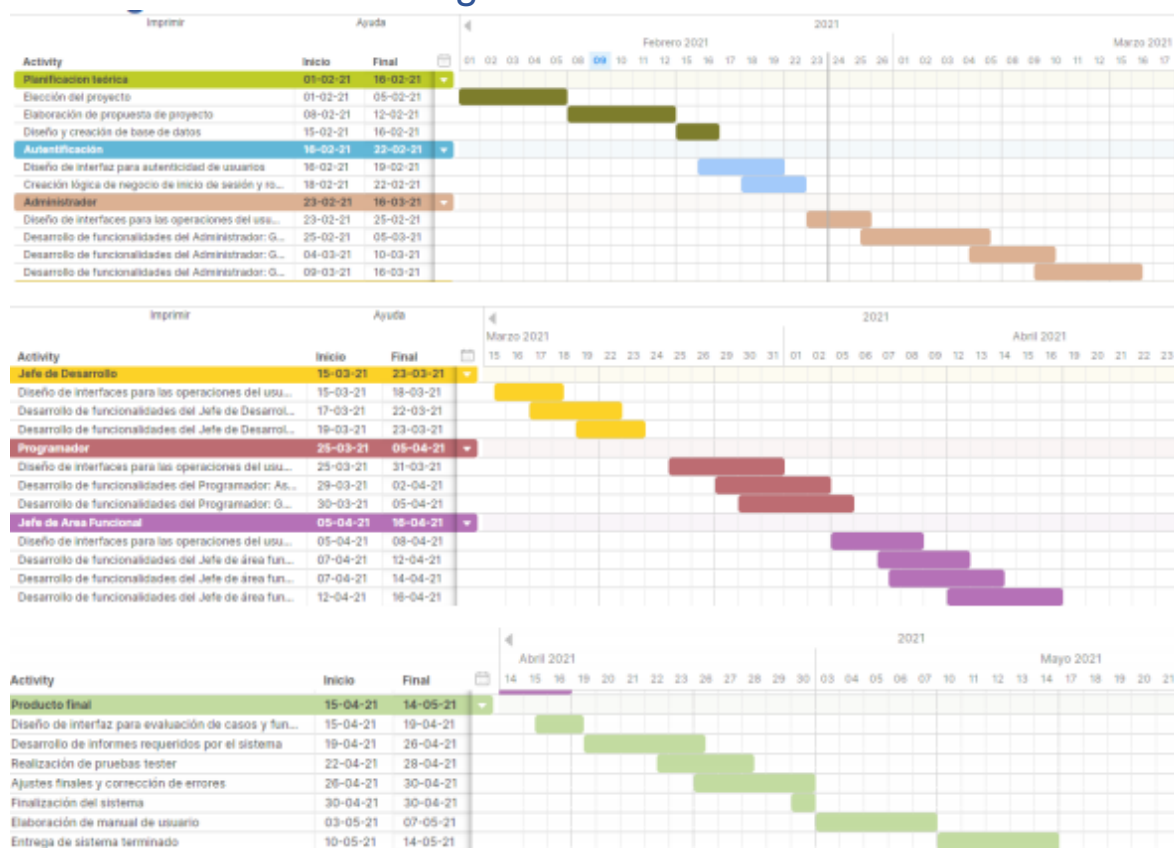
Tabla Casos

[illegible]

Tabla Bitácora

LAPTOP-1BPNONU4\...O - dbo.bitacora				LAPTOP-1BPNONU4...o_POO - dbo.caso			
	id	idCaso	descripcion				
▶	1	DF30102020851	Añadido login				
	2	DF30102020851	Añadido web s...				
	3	DF852021846	Creación del lo...				
	4	DF852021846	Proyecto termi...				
	5	DF852021846	Control de usu...				
	1003	DF852021650	Creación de we...				
	1004	DF852021650	Creación de co...				
	1005	DF852021650	Identificación d...				
	1006	DF852021821	Modificación d...				
	1007	DF852021821	Creación de m...				
	1008	DF852021821	Mejora de servi...				
	1009	DF852021821	Identificación d...				
✱	NULL	NULL	NULL				

Cronograma de actividades



Conclusión

El programa desarrollado está orientado a mejorar la gestión de las necesidades de soluciones digitales dentro de una empresa sin importar su tamaño, ya que puede manejar una gran variedad de roles dentro del sistema, sin mencionar que se pueden agregar nuevos usuarios como sean requerido dentro de los departamentos, los cuales también pueden ser creados según las necesidades de cada empresa cruce.

La manera de como están divididas las responsabilidades dentro del sistema hace que el seguimiento de las solicitudes y casos sea más eficiente, ya que cada usuario se encarga de sus propias propuestas y como se van desarrollando hasta estar terminadas.

Nuestro programa está pensado para estar en una conexión LAN de una empresa con una estructura definida por el documento, donde detallan los roles y las acciones que cumple cada empleado. Con lo cual podemos decir que los roles e interacciones entre estos ya han sido definidos en documentos anteriores, por lo que adiciones de cualquier naturaleza, deben ser estudiadas y consideraras por los miembros del proyecto así también las revisiones a futuro están contempladas dentro del proyecto así proporcionando nuevas características o arreglos que se le puedan hacer al sistema cuando se ha implementado y se hagan pruebas de su rendimiento ya en su uso cotidiano.

Bibliografía

- A. Aho, J. H. (1987). *Data Structures and Algorithms*. Addison Wesley.
- Gutiérrez, F. (2001). *Estructuras de Datos. Especificación, Diseño e Implementación*. Ed. Edicions UPC.
- Joyanes Aguilar, L. (1987). *Metodología de la programación: diagramas de flujo algoritmos y programación estructurada*. México: McGraw Hill.
- Joyanes Aguilar, L. (2003). *Fundamentos de programación : algoritmos y estructura de datos y objetos*. Madrid : McGraw-Hill.
- Julien Bascha, L. J. (1999). *Journal of Algorithms*. Paderborn: Elsevier.
- Loomis S, M. E. (1991). *Estructura de datos y organización de archivos*. México: Prentice-Hall Hispanoamericana.