Universidad Mariano Gálvez de Guatemala

Escuela Nacional Central de Agricultura

–ENCA-

Facultad de Ingeniería en Sistemas y Ciencias de la Computación Ejercicio Profesional Supervisado –EPS-

Informe de Resultados para la Escuela Nacional Central de Agricultura –ENCA-, Bajo Subvención y Programación de Desembolsos.

Periodo: ENERO-2024

f. _____

EPSista. Briam Garcia Gómez.

f. _____

Ing. Gloria Ercira Colindres Solórzano.

Encargada de la unidad de Informática.

f. _____

Ing. Jorge Escobar

Sub Director ENCA.

Informe de Avances Sistema de Recursos Humanos.

ENERO-2024

EPSista. Briam Garcia Gómez.

Índice.

Introducción. 4

Creación de tablas para terminar solicitud de Persona. 5

Creación de Procedimientos almacenados de las tablas finales del expediente de solicitud de persona. 10

Capa de datos. 19

Capa de negocio. 32

Capa de Vista. 36

Lógica de funcionamiento de la solicitud de persona de ingreso de todos sus datos para crear solicitud de empleo. 38

Compilación de Aplicación y funcionamiento de ingreso de solicitud completa de datos de una persona. 49

Conclusión. 62

Referencias. 63

Introducción.

En este Avance, se abordará la creación de tablas y procedimientos esenciales para gestionar datos relacionados con solicitudes de empleo. A través de una estructura de programación en capas en C#, exploraremos la lógica detrás de la aplicación, destacando la eficiencia y modularidad del código. La ejecución de la aplicación permitirá un ingreso de datos fluido para la solicitud de empleo, brindando una experiencia intuitiva y efectiva.

Creación de tablas para terminar solicitud de Persona.

Para la creación de las ultimas tablas de la solicitud de persona veremos el diagrama anterior del informe 2 donde veremos las tablas que llevamos al momento y así poder crear las ultimas tablas de la solicitud de empleo de una persona.

Diagrama versión 2 de base de datos "enca".

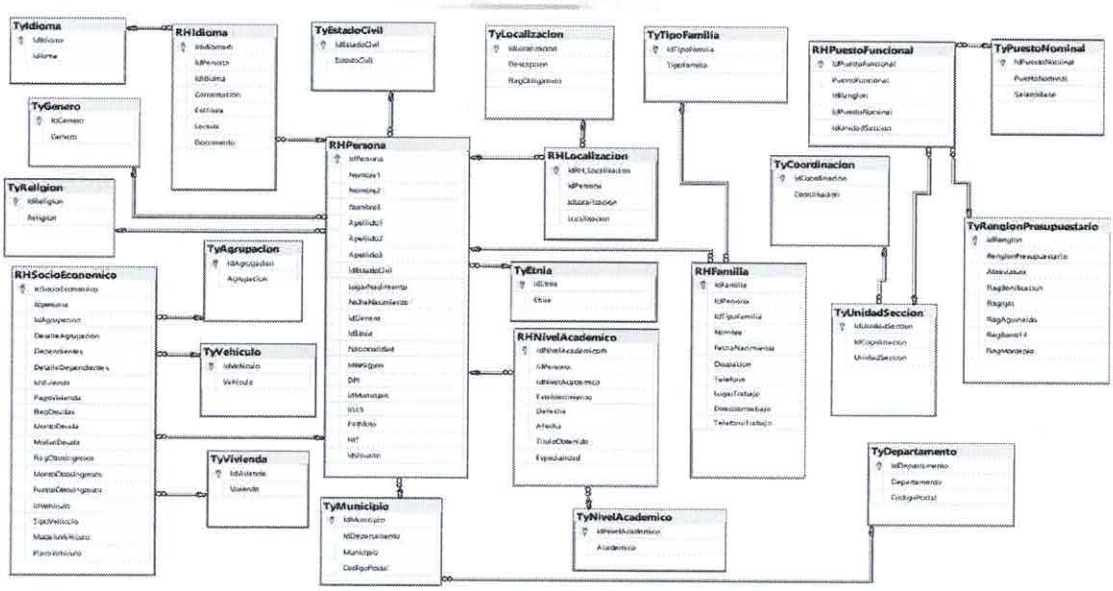
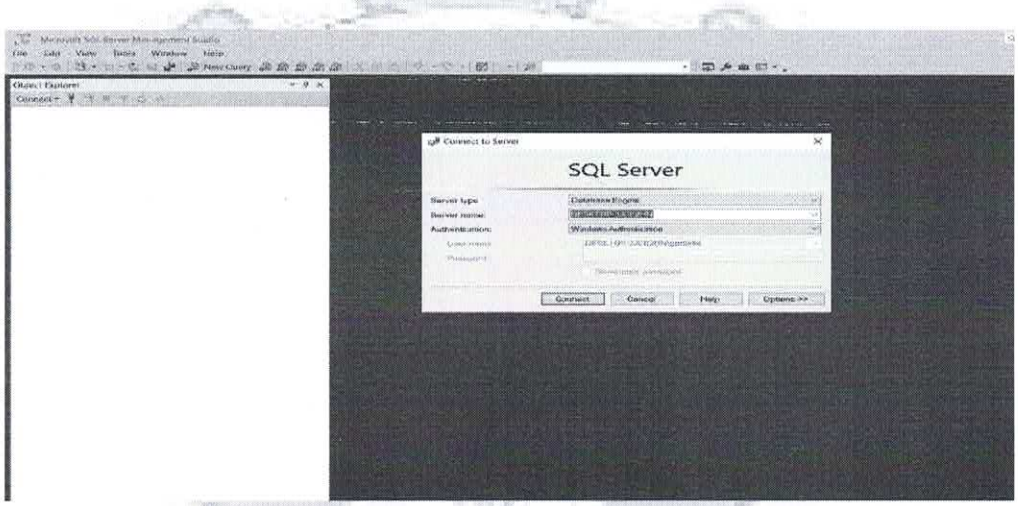


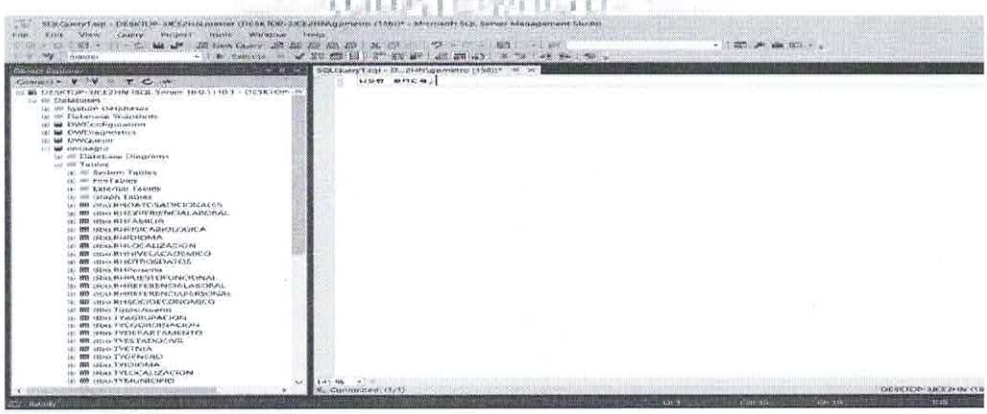
DIAGRAMA ER FASE_2 BASE DE DATOS ENCA

Este es el diagrama de la versión 2 de nuestra base de datos, pero viendo la solicitud de empleo para una persona nos faltan las siguientes tablas Experiencia laboral, Información física biológica, Referencias laborales, Referencias Personales, Otros datos y datos Adicionales de una persona en base a la solicitud de empleo. Para ello crearemos estas tablas con la relación lógica que tiene con nuestra tabla maestra que es RHPersona.

Para ello vamos a nuestra base de datos al SQL SERVER.



Nos conectamos a nuestro servidor.



Ya teniendo nuestra conexión establecida y ejecutado un NEW QUERY aquí vamos a crear los códigos SQL para creación de las tablas antes mencionadas con sus relaciones lógicas.

Código de creación de tabla RHExperiencialaboral.

```
create table RHEXPERIENCIALABORAL(
IDEXPERIENCIA int primary key identity(1,1),
IdPersona int,
EMPRESA varchar(100),
TELEFONO varchar(20),
JEFE varchar(100),
PUESTO varchar(100),
SALARIO VARCHAR(30),
FECHA_INGRESO date,
FECHA_RETIRO date,
MOTIVO_RETIRO varchar(255),
REFERENCIAS varchar(20),
DESCRIPCION varchar(255),
CONSTRAINT FK_RHEXPERIENCIALABORAL_RHPersona FOREIGN KEY (IdPersona)
        REFERENCES RHPersona(IdPersona));
```

Código de creación de tabla RHFisicaBiologica.

```
create table RHFISICABIOLOGICA(
IDFISICABIO int primary key identity(1,1),
IdPersona int,
ENFERMEDAD varchar(200),
DIABETES varchar(20),
ACCIDENTE varchar(50),
OPERACION varchar(20),
ALERGIAS varchar(20),
TRATAMIENTO varchar(10),
ESPECIFIQUE varchar(255),
LENTES varchar(10),
AUDITIVO varchar(10),
DISCAPACIDAD varchar(100),
DROGRAS varchar(10),
ALCOHOL varchar(10),
FUMA varchar(10),
PESO varchar(15),
ESTATURA varchar(20),
SANGRE varchar(20),
PASATIEMPOS varchar(255),
DEPORTES varchar(255),
CONSTRAINT FK_RHFISICABIOLOGICA_RHPersona FOREIGN KEY (IdPersona)
        REFERENCES RHPersona(IdPersona));
```

Código de creación de tabla RHReferencialaborl.

```
create table RHREFERENCIALABORAL(
IDREFLABORAL int primary key identity(1,1),
IdPersona int,
EMPRESA varchar(100),
TELEFONO varchar(20),
RELACION varchar(255),
CONSTRAINT FK_RHREFERENCIALABORAL_RHPersona FOREIGN KEY (IdPersona)
        REFERENCES RHPersona(IdPersona));
```

Código de creación de tabla RHReferenciapersonal.

```
create table RHREFERENCIAPERSONAL(
IDREFPERSONAL int primary key identity(1,1),
IdPersona int,
NOMBRE varchar(100),
TELEFONO varchar(20),
RELACION varchar(255),
CONSTRAINT FK_RHREFERENCIAPERSONAL_RHPersona FOREIGN KEY (IdPersona)
        REFERENCES RHPersona(IdPersona));
```

Código de creación de tabla RHOtrosdatoos.

```
create table RHOTROSDATOS(
IDOTROS int primary key identity(1,1),
IdPersona int,
TRABAJOENCA varchar(100),
FECHAT date,
PUESTO varchar(100),
SOLICITUDENCA varchar(100),
FECHAS date,
PLAZA varchar(100),
DISPONIBILIDAD varchar(100),
FAMILIAR_ENCA varchar(100),
FAMILIAR_ENCA_ACTUAL varchar(100),
ENCA_RELACION varchar(100),
PUESTO_CONOCIDO varchar(100),
PLAZA_VACANTE varchar(255),
CONSTRAINT FK_RHOTROSDATOS_RHPersona FOREIGN KEY (IdPersona)
REFERENCES RHPersona(IdPersona));
```

Código de creación de tabla RHDatosadicionales.

```
create table RHDATOSADICIONALES(
IDDATOSADI int primary key identity(1,1),
IdPersona int,
EMERGENCIAS varchar(255),
PARENTESCO varchar(100),
TELEFONO varchar(20),
CONSTRAINT FK_RHDATOSADICIONALES_RHPersona FOREIGN KEY (IdPersona)
```

```
REFERENCES RHPersona(IdPersona));
```


```
IdPersona int,
TRABAJOENCA varchar(100),
FECHAT date,
PUESTO varchar(100),
SOLICITUDENCA varchar(100),
FECHAS date,
PLAZA varchar(100),
DISPONIBILIDAD varchar(100),
FAMILIAR_ENCA varchar(100),
FAMILIAR_ENCA_ACTUAL varchar(100),
ENCA_RELACION varchar(100),
PUESTO_CONOCIDO varchar(100),
PLAZA_VACANTE varchar(255),
CONSTRAINT FK_RHOTROSDATOS_RHPersona FOREIGN KEY (IdPersona)
REFERENCES RHPersona(IdPersona));

create table RHDATOSADICIONALES(
IDDATOSADI int primary key identity(1,1),
IdPersona int,
EMERGENCIAS varchar(255),
PARENTESCO varchar(100),
TELEFONO varchar(20),
CONSTRAINT FK_RHDATOSADICIONALES_RHPersona FOREIGN KEY (IdPersona)
REFERENCES RHPersona(IdPersona));
```

ºCon estas tablas ya creadas y lógicamente relacionadas crearemos un diagrama entidad relación ER. Para ver la lógica de nuestra base de datos y así poder guiarnos de una mejor manera al programar nuestras capas en la aplicación.

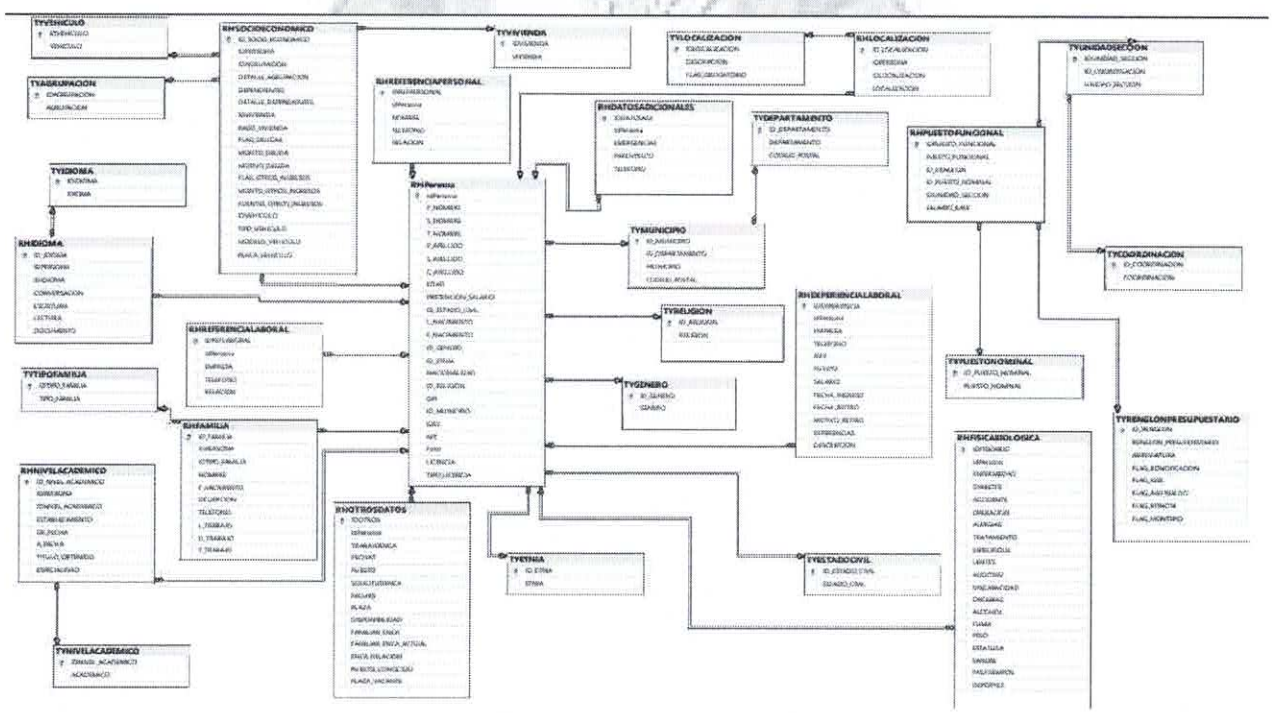


Diagrama ER final de base de datos "enca" llenado de datos de solicitud.

Con este diagrama lo que nos falta es crear los procedimientos almacenados de las tablas que se crearon anteriormente para poder realizar en nuestra aplicación en visual estudio lo que es la programación de las capas y así poder realizar una aplicación robusta y eficiente.

Creación de Procedimientos almacenados de las tablas finales del expediente de solicitud de persona.

Procedimientos almacenados de tabla RHEXPERIENCIALABORAL.

Actualizar.

```
USE [enca]
GO
/****** Object:  StoredProcedure [dbo].[sp_ActualizarRHEXPERIENCIALABORAL]    Script Date:
1/28/2024 9:12:48 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
Créate  PROCEDURE [dbo].[sp_ActualizarRHEXPERIENCIALABORAL]
   @Id INT,
        @IdPersona Int,
   @Empresa VARCHAR(100),
   @Telefono VARCHAR(20),
   @Jefe VARCHAR(100),
   @Puesto VARCHAR(100),
   @Salario VARCHAR(30),
   @Fecha_Ingreso DATE,
   @Fecha_Retiro DATE,
   @Motivo_Retiro VARCHAR(225),
   @Referencias VARCHAR(20),
   @Descripcion VARCHAR(225)
AS
BEGIN
   UPDATE RHEXPERIENCIALABORAL
   SET
          IdPersona = @IdPersona,
      EMPRESA = @Empresa,
      TELEFONO = @Telefono,
      JEFE = @Jefe,
      PUESTO = @Puesto,
      SALARIO = @Salario,
      FECHA_INGRESO = @Fecha_Ingreso,
      FECHA_RETIRO = @Fecha_Retiro,
      MOTIVO_RETIRO = @Motivo_Retiro,
```

```
    REFERENCIAS = @Referencias,
    DESCRIPCION = @Descripcion
  WHERE
    IDEXPERIENCIA = @Id;
      END;
```

Búsqueda.

```
USE [enca]
GO
/****** Object:  StoredProcedure [dbo].[sp_BuscarRHEXPERIENCIALABORAL]    Script Date: 1/28/2024
9:13:23 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
Create procedure [dbo].[sp_BuscarRHEXPERIENCIALABORAL]
@Persona varchar(200)
as
begin
select IDEXPERIENCIA as Id,
    CONCAT(P.P_APELLIDO,P.S_NOMBRE,P.P_APELLIDO,P.S_APELLIDO) as Persona,
        E.EMPRESA,
        E.TELEFONO,
        E.JEFE,
        E.PUESTO,
        E.SALARIO,
        E.FECHA_INGRESO,
        E.FECHA_INGRESO,
        E.MOTIVO_RETIRO,
        E.REFERENCIAS,
        E.DESCRIPCION
from RHEXPERIENCIALABORAL E
inner join RHPersona P on P.IdPersona = E.IdPersona
where CONCAT(P.P_APELLIDO,' ',P.S_NOMBRE,' ',P.P_APELLIDO,' ',P.S_APELLIDO) like '%' +
@Persona + '%';
      end;
```

Insertar.

```
USE [enca]
GO
/****** Object:  StoredProcedure [dbo].[sp_InsertarRHEXPERIENCIALABORAL]    Script Date: 1/28/2024
9:15:05 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
Create PROCEDURE [dbo].[sp_InsertarRHEXPERIENCIALABORAL]
  @IdPersona INT,
  @Empresa VARCHAR(100),
  @Telefono VARCHAR(20),
  @Jefe VARCHAR(100),
```

```
    @Puesto VARCHAR(100),
    @Salario VARCHAR(30),
    @Fecha_Ingreso DATE,
    @Fecha_Retiro DATE,
    @Motivo_Retiro VARCHAR(225),
    @Referencias VARCHAR(20),
    @Descripcion VARCHAR(225)
AS
BEGIN
    INSERT INTO RHEXPERIENCIALABORAL (IdPersona, EMPRESA, TELEFONO, JEFE, PUESTO,
SALARIO, FECHA_INGRESO, FECHA_RETIRO,
                                MOTIVO_RETIRO, REFERENCIAS, DESCRIPCION)
    VALUES (@IdPersona, @Empresa, @Telefono, @Jefe, @Puesto, @Salario, @Fecha_Ingreso,
@Fecha_Retiro, @Motivo_Retiro,
        @Referencias, @Descripcion);
END;
```

Listar.

```
USE [enca]
GO
/****** Object:  StoredProcedure [dbo].[sp_ListarHomeRHEXPERIENCIALABORAL]    Script Date:
1/28/2024 9:15:37 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[sp_ListarHomeRHEXPERIENCIALABORAL]
@IdExperienciaLaboral int
as
begin
select
    E.EMPRESA,
            E.TELEFONO,
            E.JEFE,
            E.PUESTO,
            E.SALARIO,
            E.FECHA_INGRESO,
            E.FECHA_INGRESO,
            E.MOTIVO_RETIRO,
            E.REFERENCIAS,
            E.DESCRIPCION
from RHEXPERIENCIALABORAL E
inner join RHPersona P on P.IdPersona = E.IdPersona
where E.IdPersona = @IdExperienciaLaboral;
end;
```

Procedimientos almacenados de tabla RHFISICABIOLOGICA.

Actualizar.

```
USE [enca]
GO
```

```
/****** Object:  StoredProcedure [dbo].[sp_ActualizarRHFISICABIOLOGICA]    Script Date: 1/28/2024
9:45:26 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Procedimiento para actualizar registros en RHFISICABIOLOGICA
Créate PROCEDURE [dbo].[sp_ActualizarRHFISICABIOLOGICA]
    @Id INT,
    @IdPersona INT,
    @Enfermedad VARCHAR(200),
    @Diabetes VARCHAR(20),
    @Accidente VARCHAR(50),
    @Operacion VARCHAR(20),
    @Alergias VARCHAR(20),
    @Tratamiento VARCHAR(10),
    @Espesifique VARCHAR(225),
    @Lentes VARCHAR(10),
    @Auditivo VARCHAR(10),
    @Discapacidad VARCHAR(100),
    @Drogas VARCHAR(10),
    @Alcohol VARCHAR(10),
    @Fuma VARCHAR(10),
    @Peso VARCHAR(15),
    @Estatura VARCHAR(20),
    @Sangre VARCHAR(20),
    @Pasatiempos VARCHAR(225),
    @Deportes VARCHAR(225)
AS
BEGIN
    UPDATE RHFISICABIOLOGICA
    SET
        IdPersona = @IdPersona,
        ENFERMEDAD = @Enfermedad,
        DIABETES = @Diabetes,
        ACCIDENTE = @Accidente,
        OPERACION = @Operacion,
        ALERGIAS = @Alergias,
        TRATAMIENTO = @Tratamiento,
        ESPECIFIQUE = @Espesifique,
        LENTES = @Lentes,
        AUDITIVO = @Auditivo,
        DISCAPACIDAD = @Discapacidad,
        DROGAS = @Drogas,
        ALCOHOL = @Alcohol,
        FUMA = @Fuma,
        PESO = @Peso,
        ESTATURA = @Estatura,
        SANGRE = @Sangre,
        PASATIEMPOS = @Pasatiempos,
        DEPORTES = @Deportes
    WHERE
        IDFISICABIO = @Id;
END;
```

Búsqueda.

```
USE [enca]
GO
/****** Object:  StoredProcedure [dbo].[sp_BuscarRHFISICABIOLOGICA]    Script Date: 1/28/2024
9:46:18 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Procedimiento para buscar registros en RHFISICABIOLOGICA por persona
Créate PROCEDURE [dbo].[sp_BuscarRHFISICABIOLOGICA]
   @Persona VARCHAR(200)
AS
BEGIN
   SELECT
         F.IDFISICABIO,
      CONCAT(P.P_APELLIDO, P.S_NOMBRE, P.P_APELLIDO, P.S_APELLIDO) AS Persona,
      F.ENFERMEDAD,
      F.DIABETES,
      F.ACCIDENTE,
      F.OPERACION,
      F.ALERGIAS,
      F.TRATAMIENTO,
      F.ESPECIFIQUE,
      F.LENTES,
      F.AUDITIVO,
      F.DISCAPACIDAD,
      F.DROGAS,
      F.ALCOHOL,
      F.FUMA,
      F.PESO,
      F.ESTATURA,
      F.SANGRE,
      F.PASATIEMPOS,
      F.DEPORTES
   FROM
      RHFISICABIOLOGICA F
      INNER JOIN RHPersona P ON P.IdPersona = F.IdPersona
   WHERE
      CONCAT(P.P_APELLIDO, ' ', P.S_NOMBRE, ' ', P.P_APELLIDO, ' ', P.S_APELLIDO) LIKE '%' +
@Persona + '%';
END;
```

Insertar.

```
USE [enca]
GO
/****** Object:  StoredProcedure [dbo].[sp_InsertarRHFISICABIOLOGICA]    Script Date: 1/28/2024
9:47:00 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Procedimiento para insertar registros en RHFISICABIOLOGICA
Create PROCEDURE [dbo].[sp_InsertarRHFISICABIOLOGICA]
```

```
    @IdPersona INT,
    @Enfermedad VARCHAR(200),
    @Diabetes VARCHAR(20),
    @Accidente VARCHAR(50),
    @Operacion VARCHAR(20),
    @Alergias VARCHAR(20),
    @Tratamiento VARCHAR(10),
    @Espesifique VARCHAR(225),
    @Lentes VARCHAR(10),
    @Auditivo VARCHAR(10),
    @Discapacidad VARCHAR(100),
    @Drogas VARCHAR(10),
    @Alcohol VARCHAR(10),
    @Fuma VARCHAR(10),
    @Peso VARCHAR(15),
    @Estatura VARCHAR(20),
    @Sangre VARCHAR(20),
    @Pasatiempos VARCHAR(225),
    @Deportes VARCHAR(225)
AS
BEGIN
    INSERT INTO RHFISICABIOLOGICA (
        IdPersona,
        ENFERMEDAD,
        DIABETES,
        ACCIDENTE,
        OPERACION,
        ALERGIAS,
        TRATAMIENTO,
        ESPECIFIQUE,
        LENTES,
        AUDITIVO,
        DISCAPACIDAD,
        DROGAS,
        ALCOHOL,
        FUMA,
        PESO,
        ESTATURA,
        SANGRE,
        PASATIEMPOS,
        DEPORTES
    )
    VALUES (
        @IdPersona,
        @Enfermedad,
        @Diabetes,
        @Accidente,
        @Operacion,
        @Alergias,
        @Tratamiento,
        @Espesifique,
        @Lentes,
        @Auditivo,
        @Discapacidad,
        @Drogas,
        @Alcohol,
```

```
    @Fuma,
    @Peso,
    @Estatura,
    @Sangre,
    @Pasatiempos,
    @Deportes
  );
END
```

Listar.

```
USE [enca]
GO
/****** Object:  StoredProcedure [dbo].[sp_ListarHomeRHFISICABIOLOGICA]    Script Date: 1/28/2024
9:47:33 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
Create procedure [dbo].[sp_ListarHomeRHFISICABIOLOGICA]
@IdFisicaBiologica int
as
begin
select F.IDFISICABIO,
    F.ENFERMEDAD,
        F.DIABETES,
        F.ACCIDENTE,
        F.OPERACION,
        F.ALERGIAS,
        F.TRATAMIENTO,
        F.ESPECIFIQUE,
        F.LENTES,
        F.AUDITIVO,
        F.DISCAPACIDAD,
        F.DROGAS,
        F.ALCOHOL,
        F.FUMA,
        F.PESO,
        F.ESTATURA,
        F.SANGRE,
        F.PASATIEMPOS,
        F.DEPORTES


from RHFISICABIOLOGICA F
inner join RHPersona P on P.IdPersona = F.IdPersona
where F.IdPersona = @IdFisicaBiologica;
end;
```

En términos generales esta es la lógica de todas las tablas que se crearon de una forma lógica y sus procedimientos. Como la tabla otros datos, datos adicionales, referencias laborales y como personales y así.

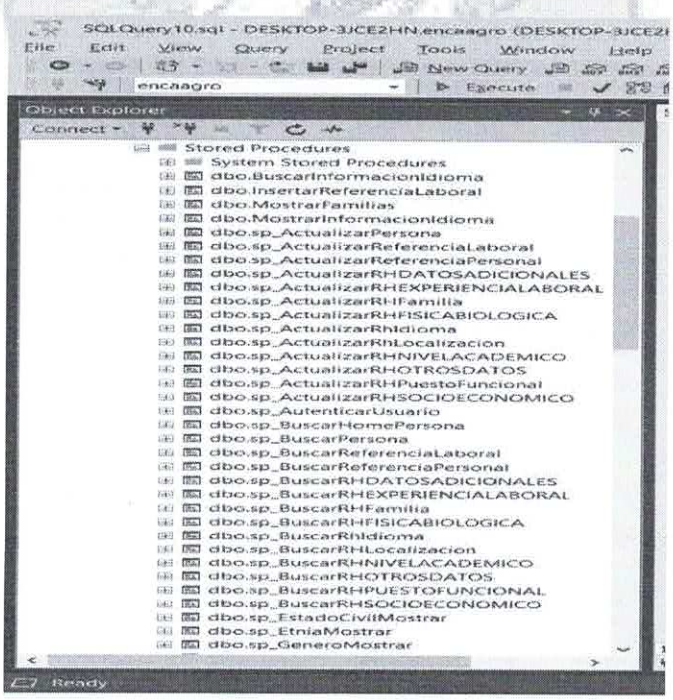
Aquí veremos todos los procedimientos almacenados de todas nuestras tablas

Con la lógica para poder hacer funcionar nuestra aplicación desde una base datos.

bien estructurada y lógicamente relacionada entres sus tablas, cabe destacar que esta programación que se le está dando a la base de datos de PL/SQL que es un estándar actual de cómo trabajar con bases de datos para sistemas transaccionales donde varios usuario están solicitando al servidor de base de datos consultas, insertando, actualizando y eliminado datos a la ves con la programación de procedimientos almacenados hacemos a nuestra base de datos robusta para este tipo de funcionalidad que va tener nuestra aplicación.


SQLQuery10.sql - DESKTOP-3JCE2HN.encaagro (DESKTOP-3JCE2H
File Edit View Query Project Tools Window Help
New Query
encaagro
Execute
Object Explorer
Connect
dbo.sp_BuscarRHSOCIOECONOMICO
dbo.sp_EstadoCivilMostrar
dbo.sp_EtniaMostrar
dbo.sp_GeneroMostrar
dbo.sp_InsertarPersona
dbo.sp_InsertarReferenciaLaboral
dbo.sp_InsertarReferenciaPersonal
dbo.sp_InsertarRHDATOSADICIONALES
dbo.sp_InsertarRHEXPERIENCIALABORAL
dbo.sp_InsertarRHFamilia
dbo.sp_InsertarRHFISICABIOLOGICA
dbo.sp_InsertarRhIdioma
dbo.sp_InsertarRHLocalizacion
dbo.sp_InsertarRHNIVELACADEMICO
dbo.sp_InsertarRHOTROSDATOS
dbo.sp_InsertarRHPuestoFuncional
dbo.sp_InsertarRHSOCIOECONOMICO
dbo.sp_ListarDepartamento
dbo.sp_listarHomeEducacion
dbo.sp_ListarHomeFamilia
dbo.sp_ListarHomeidioma
dbo.sp_ListarHomePersona
dbo.sp_ListarHomeReferenciaLaboral
dbo.sp_ListarHomeReferenciaPersonal
dbo.sp_ListarHomeRHDATOSADICIONALES
dbo.sp_ListarHomeRHEXPERIENCIALABORAL
dbo.sp_ListarHomeRHFISICABIOLOGICA
dbo.sp_ListarHomeRHOTROSDATOS
dbo.sp_ListarHomeRHSOCIOECONOMICO
dbo.sp_ListarHomeUbicacion
dbo.sp_ListarIdioma
dbo.sp_ListarIdioma_Idioma
dbo.sp_ListarMunicipio
dbo.sp_ListarMunicipioPersona
dbo.sp_ListarPersona
dbo.sp_ListarPersonaldioma
dbo.sp_ListarPuestoNominal
Ready


SQLQuery10.sql - DESKTOP-3JCE2HN.encaagro (DESKTOP-3JCE2HN\gpmetro
File Edit View Query Project Tools Window Help
New Query
encaagro
Execute
Object Explorer
Connect
dbo.sp_ListarIdioma
dbo.sp_ListarIdioma_Idioma
dbo.sp_ListarMunicipio
dbo.sp_ListarMunicipioPersona
dbo.sp_ListarPersona
dbo.sp_ListarPersonaldioma
dbo.sp_ListarPuestoNominal
dbo.sp_ListarRHEXPERIENCIALABORAL
dbo.sp_ListarRHFISICABIOLOGICA
dbo.sp_ListarRHOTROSDATOS
dbo.sp_ListarTipoFamilia
dbo.sp_ListarTYAGRUPACION
dbo.sp_ListarTyLocalizacion
dbo.sp_ListarTYNivelAcademico
dbo.sp_ListarTYRENGLONPRESUPUESTARIO
dbo.sp_ListarTYVEHICULO
dbo.sp_ListarTYVIVIENDA
dbo.sp_ListarUnidadSeccion
dbo.sp_ListRegionPresupuestario
dbo.sp_MostrarReferenciaLaboral
dbo.sp_MostrarReferenciaPersonal
dbo.sp_MostrarRHDATOSADICIONALES
dbo.sp_MostrarRHFamilia
dbo.sp_MostrarRHLocalizacion
dbo.sp_MostrarRHNIVELACADEMICO
dbo.sp_MostrarRHPersona
dbo.sp_MostrarRHPUESTOFUNCIONAL
dbo.sp_MostrarRHSOCIOECONOMICO
dbo.sp_PuestoFuncionalMostrar
dbo.sp_PuestoNominalBuscar
dbo.sp_PuestoNominalEditar
dbo.sp_PuestoNominalInsertar
dbo.sp_PuestoNominalMostrar
dbo.sp_ReligionMostrar
Functions
Database Triggers
Assemblies
141 %
Connected.
Ready


SQLQuery10.sql - DESKTOP-3JCE2HN.encaagro (DESKTOP-3JCE2H
File Edit View Query Project Tools Window Help
New Query
encaagro
Execute
Object Explorer
Connect
Stored Procedures
System Stored Procedures
dbo.BuscarInformacionIdioma
dbo.InsertarReferenciaLaboral
dbo.MostrarFamilias
dbo.MostrarInformacionIdioma
dbo.sp_ActualizarPersona
dbo.sp_ActualizarReferenciaLaboral
dbo.sp_ActualizarReferenciaPersonal
dbo.sp_ActualizarRHDATOSADICIONALES
dbo.sp_ActualizarRHEXPERIENCIALABORAL
dbo.sp_ActualizarRHFamilia
dbo.sp_ActualizarRHFISICABIOLOGICA
dbo.sp_ActualizarRhIdioma
dbo.sp_ActualizarRhLocalizacion
dbo.sp_ActualizarRHNIVELACADEMICO
dbo.sp_ActualizarRHOTROSDATOS
dbo.sp_ActualizarRHPuestoFuncional
dbo.sp_ActualizarRHSOCIOECONOMICO
dbo.sp_AutenticarUsuario
dbo.sp_BuscarHomePersona
dbo.sp_BuscarPersona
dbo.sp_BuscarReferenciaLaboral
dbo.sp_BuscarReferenciaPersonal
dbo.sp_BuscarRHDATOSADICIONALES
dbo.sp_BuscarRHEXPERIENCIALABORAL
dbo.sp_BuscarRHFamilia
dbo.sp_BuscarRHFISICABIOLOGICA
dbo.sp_BuscarRhIdioma
dbo.sp_BuscarRHLocalizacion
dbo.sp_BuscarRHNIVELACADEMICO
dbo.sp_BuscarRHOTROSDATOS
dbo.sp_BuscarRHPUESTOFUNCIONAL
dbo.sp_BuscarRHSOCIOECONOMICO
dbo.sp_EstadoCivilMostrar
dbo.sp_EtniaMostrar
dbo.sp_GeneroMostrar
Ready

Programación de las capas de la aplicación para terminar las tablas de solicitud de persona.

Ya teniendo nuestras tablas y procedimientos de cada tabla crearemos las nuevas ventanas para estas tablas en nuestra aplicación. Con la estructura de tres capas. Primero vamos a nuestra capa de datos.

Capa de datos.

Código de nuestra clase de capa de datos. Aquí se crean los datos sus métodos get y set y el constructor con datos y vacio y los métodos de insertar, actualizar, seleccionar, búsqueda y eliminar llamando a los procedimientos almacenados que se programaron en base a cada tabla.

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Capa_Datos
{
    public class cExperienciaLaboral
    {
        // Propiedades
        public int IDEXPERIENCIA { get; set; }
        public int IdPersona { get; set; }
        public string EMPRESA { get; set; }
        public string TELEFONO { get; set; }
        public string JEFE { get; set; }
        public string PUESTO { get; set; }
        public string SALARIO { get; set; }
        public DateTime FECHA_INGRESO { get; set; }
        public DateTime FECHA_RETIRO { get; set; }
        public string MOTIVO_RETIRO { get; set; }
        public string REFERENCIAS { get; set; }
        public string DESCRIPCION { get; set; }
```

```
public string Persona { get; set; }

// Constructor vacío
public cExperienciaLaboral()
{
    // Puedes inicializar propiedades predeterminadas aquí si es necesario
}

// Constructor con datos
public cExperienciaLaboral(int idPersona, string empresa, string telefono, string jefe, string puesto,
    string salario, DateTime fechaIngreso, DateTime fechaRetiro, string motivoRetiro,
    string referencias, string descripcion, string persona)
{
    IdPersona = idPersona;
    EMPRESA = empresa;
    TELEFONO = telefono;
    JEFE = jefe;
    PUESTO = puesto;
    SALARIO = salario;
    FECHA_INGRESO = fechaIngreso;
    FECHA_RETIRO = fechaRetiro;
    MOTIVO_RETIRO = motivoRetiro;
    REFERENCIAS = referencias;
    DESCRIPCION = descripcion;
    Persona = persona;
}

public string Insertar(cExperienciaLaboral experiencia)
{
    string rpta = "";
    using (SqlConnection sqlcon = new SqlConnection(Conexion.Cn))
    {
        try
        {
            sqlcon.Open();

            using (SqlCommand cmd = new SqlCommand("sp_InsertarRHEXPERIENCIALABORAL",
sqlcon))
            {
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@IdPersona", experiencia.IdPersona);
                cmd.Parameters.AddWithValue("@Empresa", experiencia.EMPRESA);
                cmd.Parameters.AddWithValue("@Telefono", experiencia.TELEFONO);
                cmd.Parameters.AddWithValue("@Jefe", experiencia.JEFE);
                cmd.Parameters.AddWithValue("@Puesto", experiencia.PUESTO);
                cmd.Parameters.AddWithValue("@Salario", experiencia.SALARIO);
                cmd.Parameters.AddWithValue("@Fecha_Ingreso", experiencia.FECHA_INGRESO);
                cmd.Parameters.AddWithValue("@Fecha_Retiro", experiencia.FECHA_RETIRO);
                cmd.Parameters.AddWithValue("@Motivo_Retiro", experiencia.MOTIVO_RETIRO);
                cmd.Parameters.AddWithValue("@Referencias", experiencia.REFERENCIAS);
                cmd.Parameters.AddWithValue("@Descripcion", experiencia.DESCRIPCION);


                rpta = cmd.ExecuteNonQuery() == 1 ? "OK" : "No ingresó el registro";
            }
        }
```

```
            catch (Exception ex)
            {
                rpta = ex.Message;
            }
        }

        return rpta;
    }

    public string Actualizar(cExperienciaLaboral experiencia)
    {
        string rpta = "";

        using (SqlConnection sqlcon = new SqlConnection(Conexion.Cn))
        {
            try
            {
                sqlcon.Open();

                using (SqlCommand cmd = new SqlCommand("sp_ActualizarRHEXPERIENCIALABORAL",
sqlcon))
                {
                    cmd.CommandType = CommandType.StoredProcedure;

                    cmd.Parameters.AddWithValue("@Id", experiencia.IDEXPERIENCIA);
                    cmd.Parameters.AddWithValue("@IdPersona", experiencia.IdPersona);
                    cmd.Parameters.AddWithValue("@Empresa", experiencia.EMPRESA);
                    cmd.Parameters.AddWithValue("@Telefono", experiencia.TELEFONO);
                    cmd.Parameters.AddWithValue("@Jefe", experiencia.JEFE);
                    cmd.Parameters.AddWithValue("@Puesto", experiencia.PUESTO);
                    cmd.Parameters.AddWithValue("@Salario", experiencia.SALARIO);
                    cmd.Parameters.AddWithValue("@Fecha_Ingreso", experiencia.FECHA_INGRESO);
                    cmd.Parameters.AddWithValue("@Fecha_Retiro", experiencia.FECHA_RETIRO);
                    cmd.Parameters.AddWithValue("@Motivo_Retiro", experiencia.MOTIVO_RETIRO);
                    cmd.Parameters.AddWithValue("@Referencias", experiencia.REFERENCIAS);
                    cmd.Parameters.AddWithValue("@Descripcion", experiencia.DESCRIPCION);
                    // Agregar parámetros para otros campos...

                    int filasAfectadas = cmd.ExecuteNonQuery();

                    rpta = filasAfectadas == 1 ? "OK" : "No se actualizó el registro 1234";

                }
            }
            catch (SqlException ex)
            {
                // Manejar la excepción y obtener detalles específicos
                foreach (SqlError error in ex.Errors)
                {
                    rpta += $"Error: {error.Message}, Número: {error.Number}, Procedimiento:
{error.Procedure}";
                    // Puedes agregar más información según tus necesidades
                }
            }
            catch (Exception ex)
```

```
        {
            rpta = ex.Message;
        }
    }

    return rpta;
}


public DataTable Mostrar()
{
    DataTable DtResultado = new DataTable("RHEXPERIENCIALABORAL");
    SqlConnection sqlconn = new SqlConnection();
    try
    {
        sqlconn.ConnectionString = Conexion.Cn;
        SqlCommand sqlcmd = new SqlCommand();
        sqlcmd.Connection = sqlconn;
        sqlcmd.CommandText = "sp_ListarRHEXPERIENCIALABORAL";
        sqlcmd.CommandType = CommandType.StoredProcedure;

        SqlDataAdapter sqlDat = new SqlDataAdapter(sqlcmd);
        sqlDat.Fill(DtResultado);
    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

public DataTable Buscar(cExperienciaLaboral experiencia)
{
    DataTable DtResultado1 = new DataTable("RHEXPERIENCIALABORAL");
    SqlConnection sqlconn = new SqlConnection();
    try
    {
        sqlconn.ConnectionString = Conexion.Cn;
        SqlCommand sqlcmd = new SqlCommand();
        sqlcmd.Connection = sqlconn;
        sqlcmd.CommandText = "sp_BuscarRHEXPERIENCIALABORAL";
        sqlcmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParBuscar = new SqlParameter();
        ParBuscar.ParameterName = "@Persona";
        ParBuscar.SqlDbType = SqlDbType.VarChar;
        ParBuscar.Size = 200;
        ParBuscar.Value = experiencia.Persona;
        sqlcmd.Parameters.Add(ParBuscar);

        SqlDataAdapter sqlDat = new SqlDataAdapter(sqlcmd);
        sqlDat.Fill(DtResultado1);
    }
    catch (Exception ex)
    {
        DtResultado1 = null;
```

```
        }
        return DtResultado1;
    }

  }
}
```


```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Capa_Datos
{
    public class cFisicoBiologicos
    {
        // Propiedades
        public int IDFISICABIO { get; set; }
        public int IdPersona { get; set; }
        public string ENFERMEDAD { get; set; }
        public string DIABETES { get; set; }
        public string ACCIDENTE { get; set; }
        public string OPERACION { get; set; }
        public string ALERGIAS { get; set; }
        public string TRATAMIENTO { get; set; }
        public string ESPECIFIQUE { get; set; }
        public string LENTES { get; set; }
```

```
        public string AUDITIVO { get; set; }
        public string DISCAPACIDAD { get; set; }
        public string DROGRAS { get; set; }
        public string ALCOHOL { get; set; }
        public string FUMA { get; set; }
        public string PESO { get; set; }
        public string ESTATURA { get; set; }
        public string SANGRE { get; set; }
        public string PASATIEMPOS { get; set; }
        public string DEPORTES { get; set; }

        public string Persona {  get; set; }

        // Constructor vacío
        public cFisicoBiologicos()
        {
            // Puedes inicializar propiedades predeterminadas aquí si es necesario
        }

        // Constructor con datos
        public cFisicoBiologicos(int idPersona, string enfermedad, string diabetes, string accidente,
            string operacion, string alergias, string tratamiento, string especifique, string lentes,
            string auditivo, string discapacidad, string drogras, string alcohol, string fuma, string
peso,
            string estatura, string sangre, string pasatiempos, string deportes, string persona)
        {
            IdPersona = idPersona;
            ENFERMEDAD = enfermedad;
            DIABETES = diabetes;
            ACCIDENTE = accidente;
```

```
        OPERACION = operacion;
        ALERGIAS = alergias;
        TRATAMIENTO = tratamiento;
        ESPECIFIQUE = especifique;
        LENTES = lentes;
        AUDITIVO = auditivo;
        DISCAPACIDAD = discapacidad;
        DROGRAS = drogras;
        ALCOHOL = alcohol;
        FUMA = fuma;
        PESO = peso;
        ESTATURA = estatura;
        SANGRE = sangre;
        PASATIEMPOS = pasatiempos;
        DEPORTES = deportes;
        Persona = persona;
    }
    public string InsertarFisicoBiologico(cFisicoBiologicos fisicoBiologicos)
    {
        string resultado = "";

        using (SqlConnection sqlcon = new SqlConnection(Conexion.Cn))
        {
            try
            {
                sqlcon.Open();

                using (SqlCommand cmd = new
SqlCommand("sp_InsertarRHFISICABIOLOGICA", sqlcon))
                {
```

```
                cmd.CommandType = CommandType.StoredProcedure;


                cmd.Parameters.AddWithValue("@IdPersona",
fisicoBiologicos.IdPersona).SqlDbType = SqlDbType.Int;

                cmd.Parameters.AddWithValue("@Enfermedad",
fisicoBiologicos.ENFERMEDAD).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Diabetes",
fisicoBiologicos.DIABETES).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Accidente",
fisicoBiologicos.ACCIDENTE).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Operacion",
fisicoBiologicos.OPERACION).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Alergias",
fisicoBiologicos.ALERGIAS).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Tratamiento",
fisicoBiologicos.TRATAMIENTO).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Espesifique",
fisicoBiologicos.ESPECIFIQUE).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Lentes",
fisicoBiologicos.LENTES).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Auditivo",
fisicoBiologicos.AUDITIVO).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Discapacidad",
fisicoBiologicos.DISCAPACIDAD).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Drogas",
fisicoBiologicos.DROGRAS).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Alcohol",
fisicoBiologicos.ALCOHOL).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Fuma", fisicoBiologicos.FUMA).SqlDbType =
SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Peso", fisicoBiologicos.PESO).SqlDbType =
SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Estatura",
fisicoBiologicos.ESTATURA).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Sangre",
fisicoBiologicos.SANGRE).SqlDbType = SqlDbType.VarChar;
```

```
                cmd.Parameters.AddWithValue("@Pasatiempos",
fisicoBiologicos.PASATIEMPOS).SqlDbType = SqlDbType.VarChar;
                cmd.Parameters.AddWithValue("@Deportes",
fisicoBiologicos.DEPORTES).SqlDbType = SqlDbType.VarChar;

                resultado = cmd.ExecuteNonQuery() == 1 ? "OK" : "No se ingresó el registro";
            }
          }
          catch (Exception ex)
          {
            resultado = ex.Message;
          }
        }

        return resultado;
    }

    public string ActualizarFisicoBiologico(cFisicoBiologicos fisicoBiologicos)
    {
        string resultado = "";

        using (SqlConnection sqlcon = new SqlConnection(Conexion.Cn))
        {
          try
          {
            sqlcon.Open();

            using (SqlCommand cmd = new
SqlCommand("sp_ActualizarRHFISICABIOLOGICA", sqlcon))
            {
              cmd.CommandType = CommandType.StoredProcedure;
```

```
                cmd.Parameters.AddWithValue("@Id",
fisicoBiologicos.IDFISICABIO).SqlDbType = SqlDbType.Int;

                cmd.Parameters.AddWithValue("@IdPersona",
fisicoBiologicos.IdPersona).SqlDbType = SqlDbType.Int;

                cmd.Parameters.AddWithValue("@Enfermedad",
fisicoBiologicos.ENFERMEDAD).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Diabetes",
fisicoBiologicos.DIABETES).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Accidente",
fisicoBiologicos.ACCIDENTE).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Operacion",
fisicoBiologicos.OPERACION).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Alergias",
fisicoBiologicos.ALERGIAS).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Tratamiento",
fisicoBiologicos.TRATAMIENTO).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Espesifique",
fisicoBiologicos.ESPECIFIQUE).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Lentes",
fisicoBiologicos.LENTES).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Auditivo",
fisicoBiologicos.AUDITIVO).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Discapacidad",
fisicoBiologicos.DISCAPACIDAD).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Drogas",
fisicoBiologicos.DROGRAS).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Alcohol",
fisicoBiologicos.ALCOHOL).SqlDbType = SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Fuma", fisicoBiologicos.FUMA).SqlDbType =
SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Peso", fisicoBiologicos.PESO).SqlDbType =
SqlDbType.VarChar;

                cmd.Parameters.AddWithValue("@Estatura",
fisicoBiologicos.ESTATURA).SqlDbType = SqlDbType.VarChar;
```

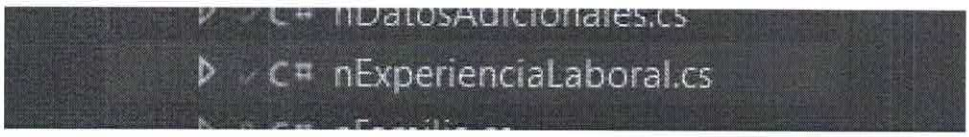
```
                cmd.Parameters.AddWithValue("@Sangre",
fisicoBiologicos.SANGRE).SqlDbType = SqlDbType.VarChar;
                cmd.Parameters.AddWithValue("@Pasatiempos",
fisicoBiologicos.PASATIEMPOS).SqlDbType = SqlDbType.VarChar;
                cmd.Parameters.AddWithValue("@Deportes",
fisicoBiologicos.DEPORTES).SqlDbType = SqlDbType.VarChar;

                int filasAfectadas = cmd.ExecuteNonQuery();

                resultado = filasAfectadas == 1 ? "OK" : "No se actualizó el registro";
            }
        }
        catch (Exception ex)
        {
            resultado = ex.Message;
        }
    }

    return resultado;
}

public DataTable Mostrar()
{
    DataTable dtResultado = new DataTable("RHFISICABIOLOGICA");

    using (SqlConnection sqlconn = new SqlConnection(Conexion.Cn))
    {
        try
        {
            sqlconn.Open();
```

```
            using (SqlCommand cmd = new SqlCommand("sp_ListarRHFISICABIOLOGICA",
sqlconn))
            {
               cmd.CommandType = CommandType.StoredProcedure;

               using (SqlDataAdapter sqlDat = new SqlDataAdapter(cmd))
               {
                  sqlDat.Fill(dtResultado);
               }
            }
         }
         catch (Exception ex)
         {
            dtResultado = null;
         }
      }

      return dtResultado;
   }

   public DataTable Buscar(cFisicoBiologicos fisicoBiologicos)
   {
      DataTable dtResultado = new DataTable("RHFISICABIOLOGICA");
      using (SqlConnection sqlconn = new SqlConnection(Conexion.Cn))
      {
         try
         {
            sqlconn.Open();

            using (SqlCommand cmd = new SqlCommand("sp_BuscarRHFISICABIOLOGICA",
sqlconn))
```

```
            {
                cmd.CommandType = CommandType.StoredProcedure;

                SqlParameter parBuscar = new SqlParameter();
                parBuscar.ParameterName = "@Persona";
                parBuscar.SqlDbType = SqlDbType.VarChar;
                parBuscar.Size = 200;
                parBuscar.Value = fisicoBiologicos.Persona;
                cmd.Parameters.Add(parBuscar);

                using (SqlDataAdapter sqlDat = new SqlDataAdapter(cmd))
                {
                    sqlDat.Fill(dtResultado);
                }
            }
        }
        catch (Exception ex)
        {
            dtResultado = null;
        }
    }
    return dtResultado;
  }
 }
}
```

En resumen, esto se hace en la capa de datos donde se crea la estructura de los datos sus métodos set y get y las funciones como la de Ingresar, Actualizar, búsqueda y selección de los datos según nuestros procedimientos almacenados.

Capa de negocio.




```
using Capa_Datos;
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Capa_Negocio
{
    public class nExperienciaLaboral
    {
        public static string Insertar(
            int idpersona, string empresa,string telefono,
            string jefe, string puesto,string salario,
            DateTime fechaIngreso,
            DateTime fechaRetiro,
            string motivoRetiro,
            string referencias,
            string descripcion)
        {
            cExperienciaLaboral Obj = new cExperienciaLaboral();
            Obj.IdPersona = idpersona;
            Obj.EMPRESA = empresa;
            Obj.TELEFONO = telefono;
            Obj.JEFE = jefe;
            Obj.SALARIO = salario;
            Obj.PUESTO = puesto;
            Obj.FECHA_INGRESO = fechaIngreso;
            Obj.FECHA_RETIRO = fechaRetiro;
            Obj.MOTIVO_RETIRO = motivoRetiro;
            Obj.REFERENCIAS = referencias;
            Obj.DESCRIPCION = descripcion;

            return Obj.Insertar(Obj);

        }

        public static string Actualizar(
            int idexperiencia,
            int idpersona,
            string empresa,
            string telefono,
            string jefe, string puesto,
            string salario,
            DateTime fechaIngreso,
            DateTime fechaRetiro,
            string motivoRetiro,
            string referencias,
```

```
            string descripcion)
        {
            cExperienciaLaboral Obj = new cExperienciaLaboral();

            Obj.IDEXPERIENCIA = idexperiencia;
            Obj.IdPersona = idpersona;
            Obj.EMPRESA = empresa;
            Obj.TELEFONO = telefono;
            Obj.JEFE = jefe;
            Obj.PUESTO = puesto;
            Obj.SALARIO = salario;
            Obj.FECHA_INGRESO = fechaIngreso;
            Obj.FECHA_RETIRO = fechaRetiro;
            Obj.MOTIVO_RETIRO = motivoRetiro;
            Obj.REFERENCIAS = referencias;
            Obj.DESCRIPCION = descripcion;

            return Obj.Actualizar(Obj);
        }

        public static DataTable Mostrar()
        {
            return new cExperienciaLaboral().Mostrar();
        }

        public static DataTable Buscar(string persona)
        {
            cExperienciaLaboral Obj = new cExperienciaLaboral();
            Obj.Persona = persona;
            return Obj.Buscar(Obj);
        }
    }
}
```

nFisicoBiologico.cs

```
using Capa_Datos;
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Capa_Negocio
{
    public class nFisicoBiologico
    {
        public static string Insertar(
            int idPersona, string enfermedad,
            string diabetes, string accidente,
            string operacion, string alergias,
            string tratamiento, string especifique,
            string lentes,string auditivo, string discapacidad,
            string drogras, string alcohol, string fuma, string peso,
```

```
            string estatura, string sangre, string pasatiempos,
            string deportes)
        {

            cFisicoBiologicos Obj = new cFisicoBiologicos();
            Obj.IdPersona = idPersona;
            Obj.ENFERMEDAD = enfermedad;
            Obj.DIABETES = diabetes;
            Obj.ACCIDENTE = accidente;
            Obj.OPERACION = operacion;
            Obj.ALERGIAS = alergias;
            Obj.TRATAMIENTO = tratamiento;
            Obj.ESPECIFIQUE = especifique;
            Obj.LENTES = lentes;
            Obj.AUDITIVO = auditivo;
            Obj.DISCAPACIDAD    = discapacidad;
            Obj.DROGRAS = drogras;
            Obj.ALCOHOL = alcohol;
            Obj.FUMA = fuma;
            Obj.PESO = peso;
            Obj.ESTATURA = estatura;
            Obj.SANGRE = sangre;
            Obj.PASATIEMPOS = pasatiempos;
            Obj.DEPORTES = deportes;
             return Obj.InsertarFisicoBiologico( Obj );

        }

        public static string Actualizar(int idFisicoBiologico, int idPersona, string
enfermedad, string diabetes, string accidente,
            string operacion, string alergias, string tratamiento, string
especifique, string lentes,
            string auditivo, string discapacidad, string drogras, string alcohol,
string fuma, string peso,
            string estatura, string sangre, string pasatiempos, string deportes)
        {
            cFisicoBiologicos Obj = new cFisicoBiologicos();
            Obj.IDFISICABIO = idFisicoBiologico;
            Obj.IdPersona = idPersona;
            Obj.ENFERMEDAD = enfermedad;
            Obj.DIABETES = diabetes;
            Obj.ACCIDENTE = accidente;
            Obj.OPERACION = operacion;
            Obj.ALERGIAS = alergias;
            Obj.TRATAMIENTO = tratamiento;
            Obj.ESPECIFIQUE = especifique;
            Obj.LENTES = lentes;
            Obj.AUDITIVO = auditivo;
            Obj.DISCAPACIDAD = discapacidad;
            Obj.DROGRAS = drogras;
            Obj.ALCOHOL = alcohol;
            Obj.FUMA = fuma;
            Obj.PESO = peso;
            Obj.ESTATURA = estatura;
            Obj.SANGRE = sangre;
            Obj.PASATIEMPOS = pasatiempos;
            Obj.DEPORTES = deportes;
            return Obj.ActualizarFisicoBiologico(Obj);
```

```
        }

        public static DataTable Buscar(string persona)
        {

            cFisicoBiologicos obj = new cFisicoBiologicos();
            obj.Persona = persona;
            return obj.Buscar(obj);
        }

        public static DataTable Mostrar()
        {
            return new cFisicoBiologicos().Mostrar();
        }

    }
}
```

C# nDatosAdicionales.cs

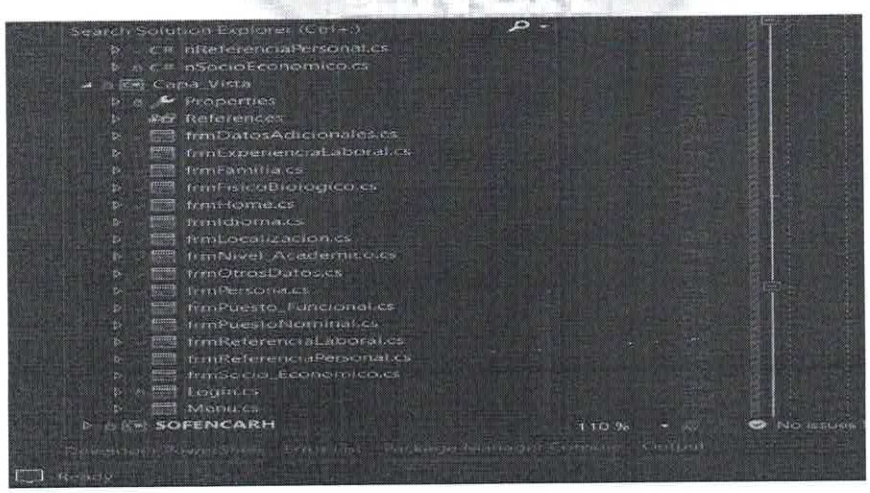
```
using Capa_Datos;
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Capa_Negocio
{
    public class nDatosAdicionales
    {

        public static string Insertar(
            int idpersona, string emergencias,
            string parentesco,string telefono
               )
        {
            cDatosAdicionales obj = new cDatosAdicionales();

            obj.IdPersona = idpersona;
            obj.EMERGENCIAS = emergencias;
            obj.PARENTESCO = parentesco;
            obj.TELEFONO = telefono;

            return obj.Insertar(obj);
        }

        public static string Actualizar(
            int id, int idPersona, string emergencias,
            string parentesco,string telefono
            )
        {
            cDatosAdicionales obj = new cDatosAdicionales();
```

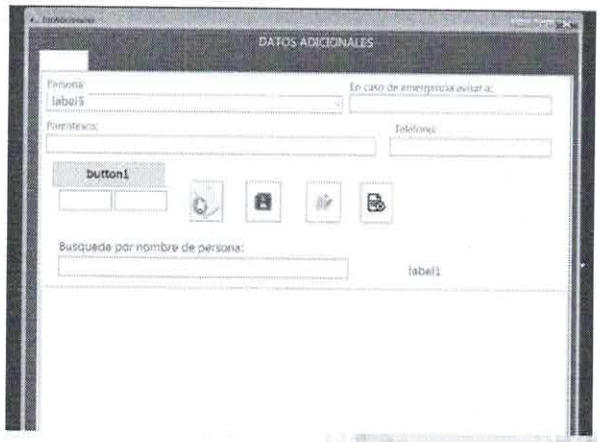
```
            obj.IDDATOSADI = id;
            obj.IdPersona = idPersona;
            obj.EMERGENCIAS = emergencias;
            obj.PARENTESCO = parentesco;
            obj.TELEFONO = telefono;

            return obj.Actualizar(obj);
        }

        public static DataTable Buscar(string persona)
        {
            cDatosAdicionales obj = new cDatosAdicionales();
            obj.Persona = persona;
            return obj.Buscar(obj);
        }

        public static DataTable Mostrar()
        {
            return new cDatosAdicionales().Mostrar();
        }
    }
}
```

En resumen, en la capa de negocio se crea la clase de la ventana y los string o cadenas de datos que se crean en la capa de datos estos para hacer uso en nuestra capa de vista donde se crea el diseño de nuestras ventanas.

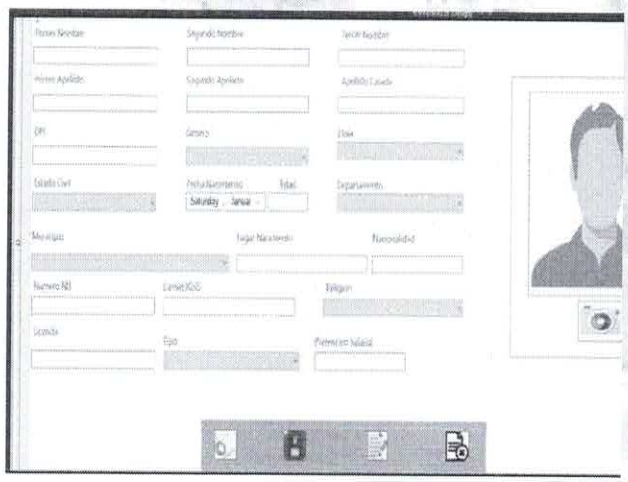
Capa de Vista.

Aquí se ven las vistas de usuarios que a continuación se ven en el siguiente tema que es lógica de funcionamiento. Ven las pantallas.




DATOS ADICIONALES
Persona:
label5
En caso de emergencia avisar a:
Parentesco:
Telefono:
Button1
Busqueda por nombre de persona:
label1


ESCUELA NACIONAL CENTRAL
DE AGRICULTURA -ENCA-
ENCA
GUATEMALA
ACCEDER
TODOS LOS DERECHOS RESERVADOS

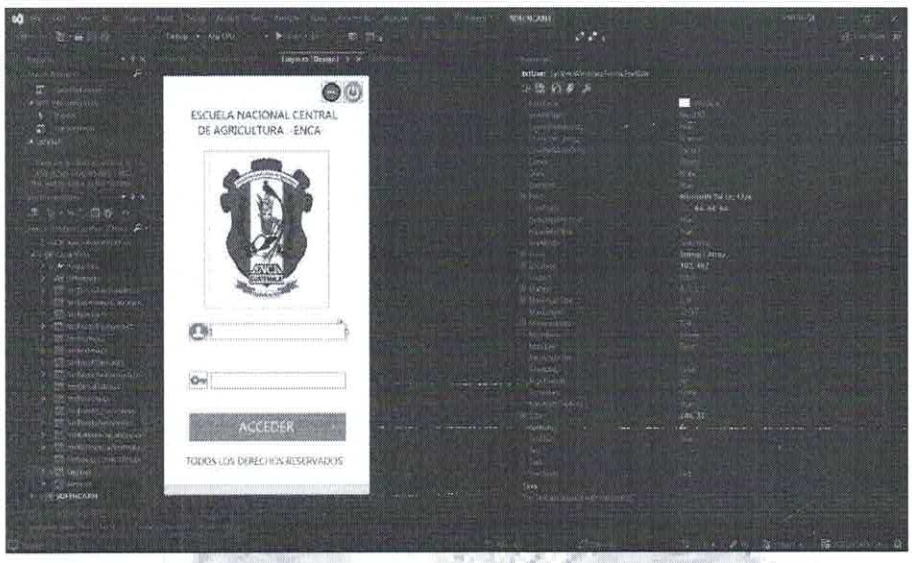



EXPERIENCIA LABORAL

Lógica de funcionamiento de la solicitud de persona de ingreso de todos sus datos para crear solicitud de empleo.

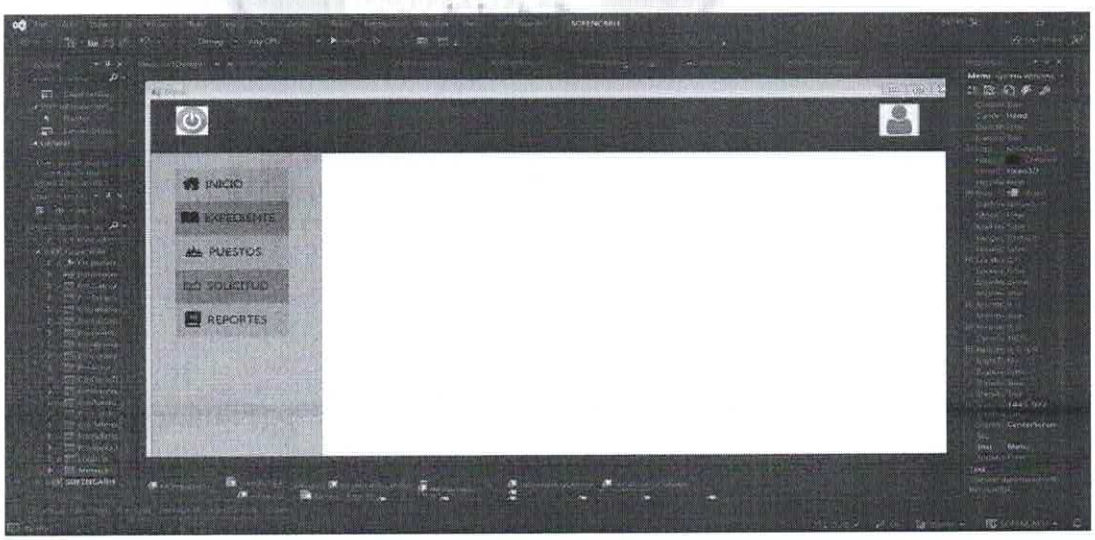
La lógica de como se estructura las pantallas de la aplicación es la siguiente.

Primero vamos a la pantalla de login donde nos pide usuario y contraseña.



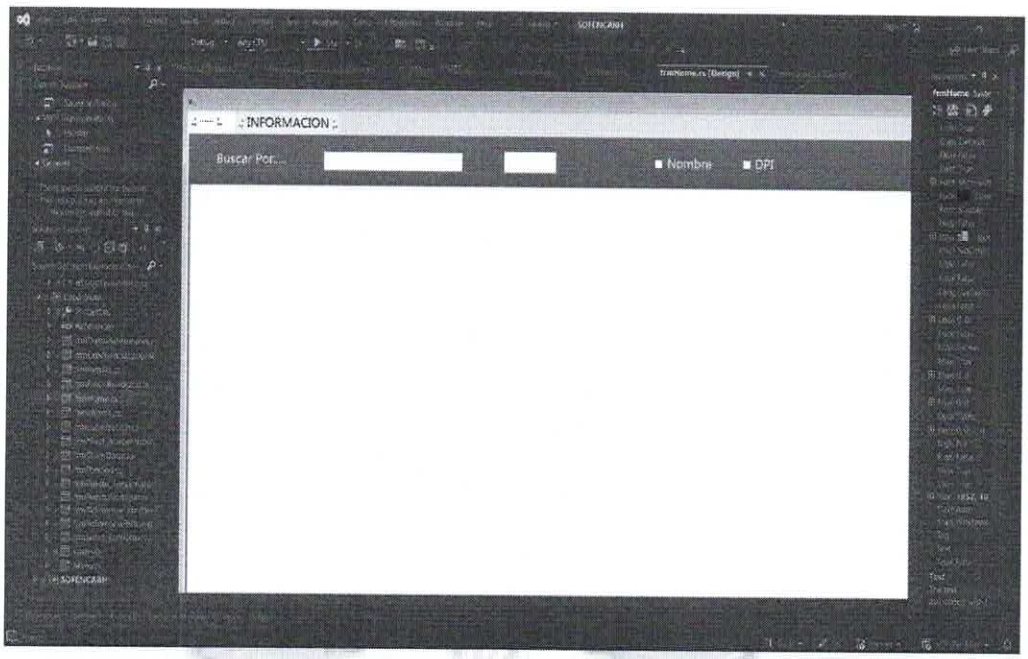
Al ingresar nuestras credenciales accedemos al menú de la aplicación.

Que es el siguiente.



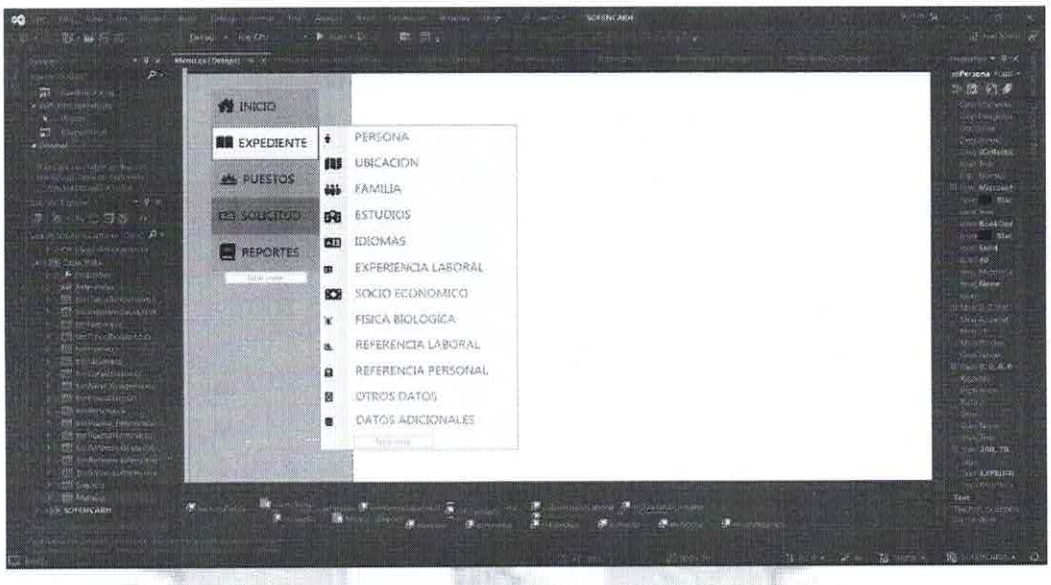
Este es menú de la aplicación en la parte izquierda tenemos nuestra barra de menú que esta compuesta por inicio, expediente, puestos, solicitudes y los reportes hasta el momento.

En inicio tendremos lo que es las personas ingresadas.



Aquí tendremos lo que son las personas en la pantalla blanca, y hay dos funciones de búsqueda de una persona que es por nombre y DPI.

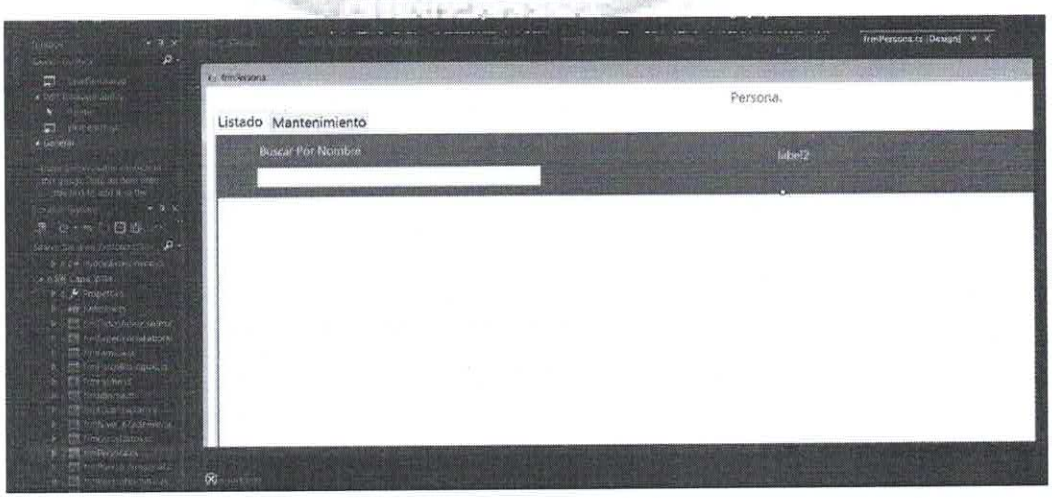
Luego regresando al menú aparece la parte de expediente donde se ingresará toda la información de la persona y la persona en sí.



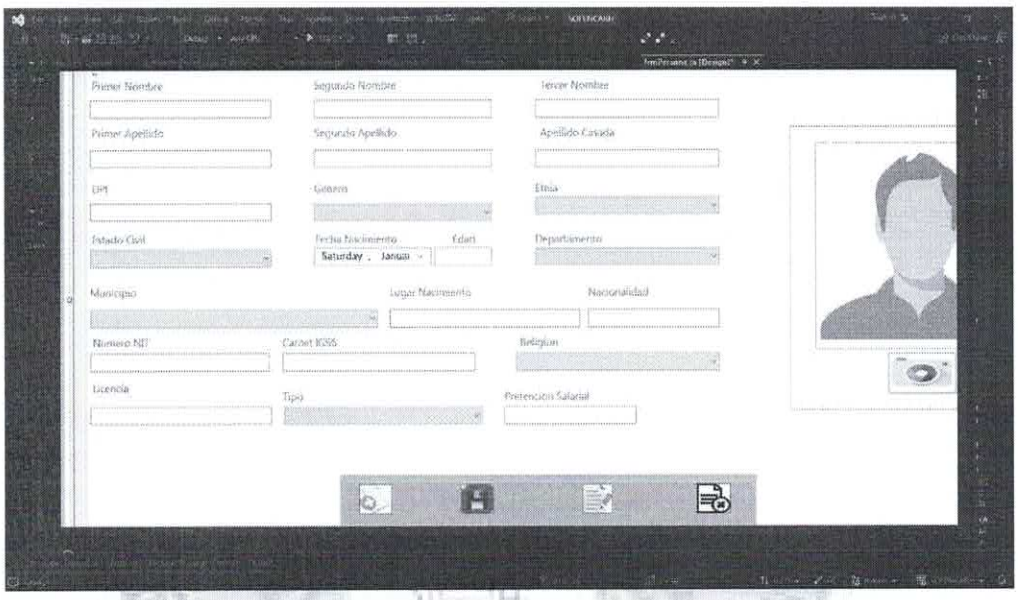
Al darle clic sobre expediente aparece un submenú donde tenemos todas las ventanas para ingresar información de datos de una persona y todos sus datos que corresponden según la solicitud de empleo de persona.

Ventana persona.

Esta ventana tiene dos funciones una donde nos muestra la lista de personas que se van ingresando al sistema.

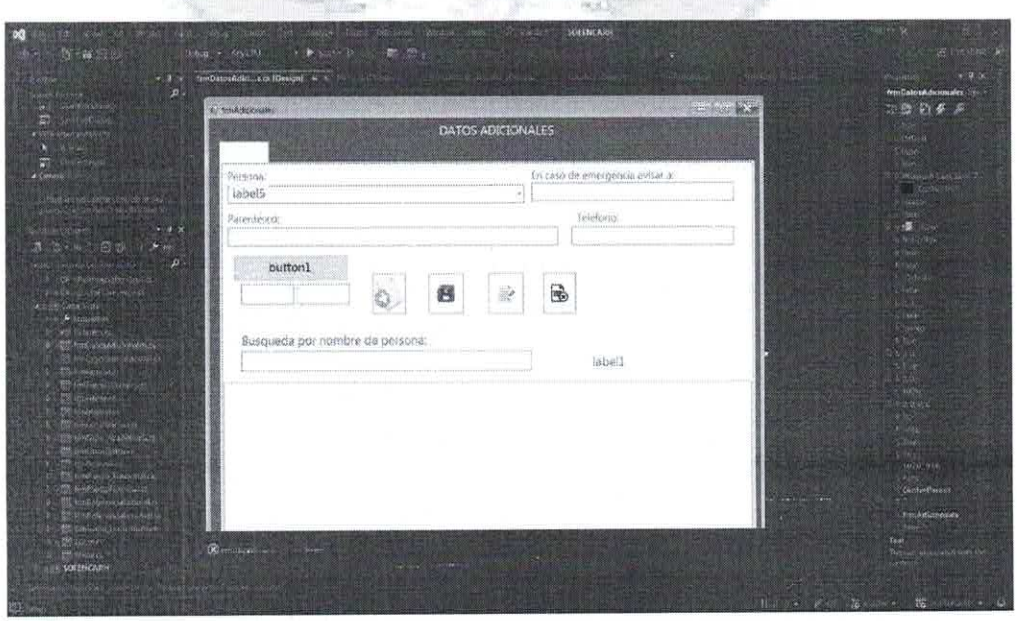


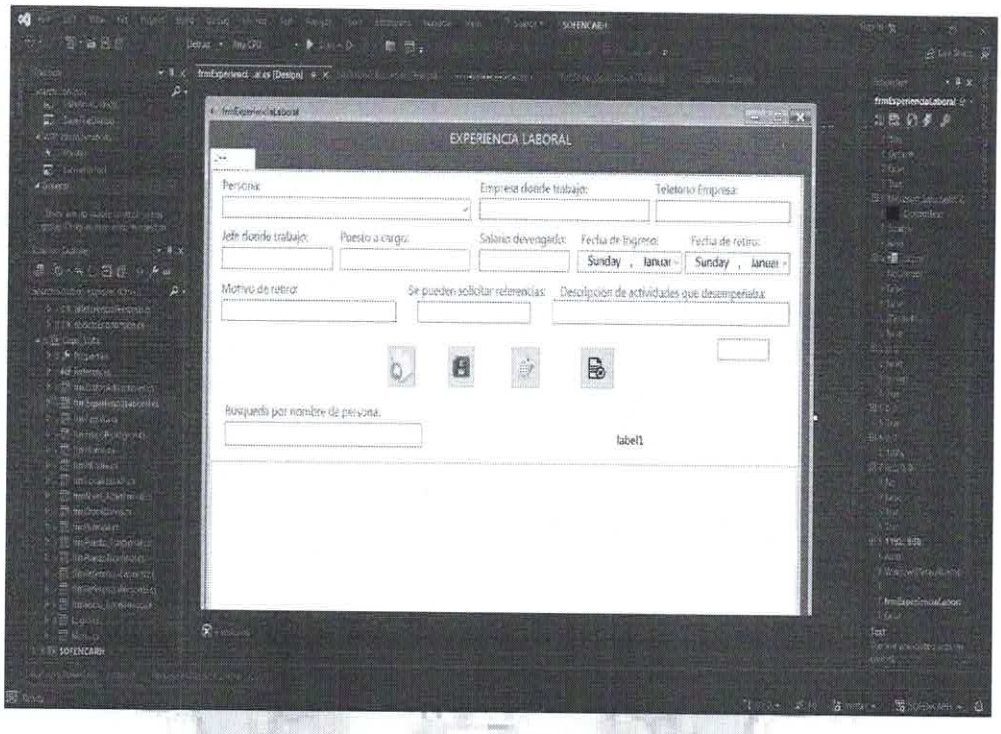
Y la parte de mantenimiento donde se ingresa una nueva persona.

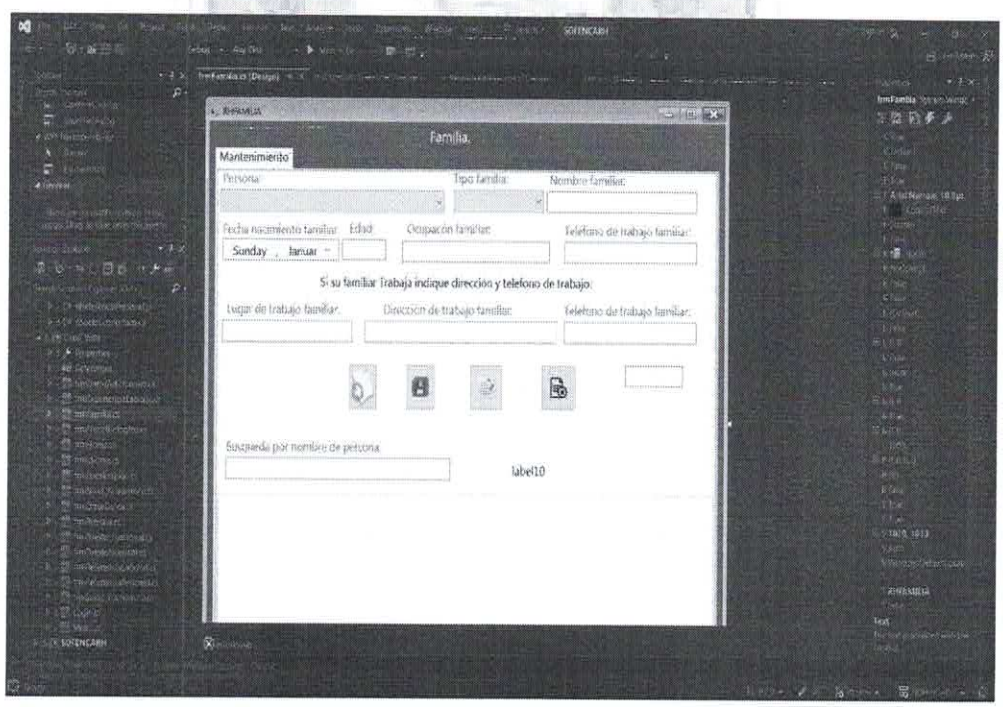


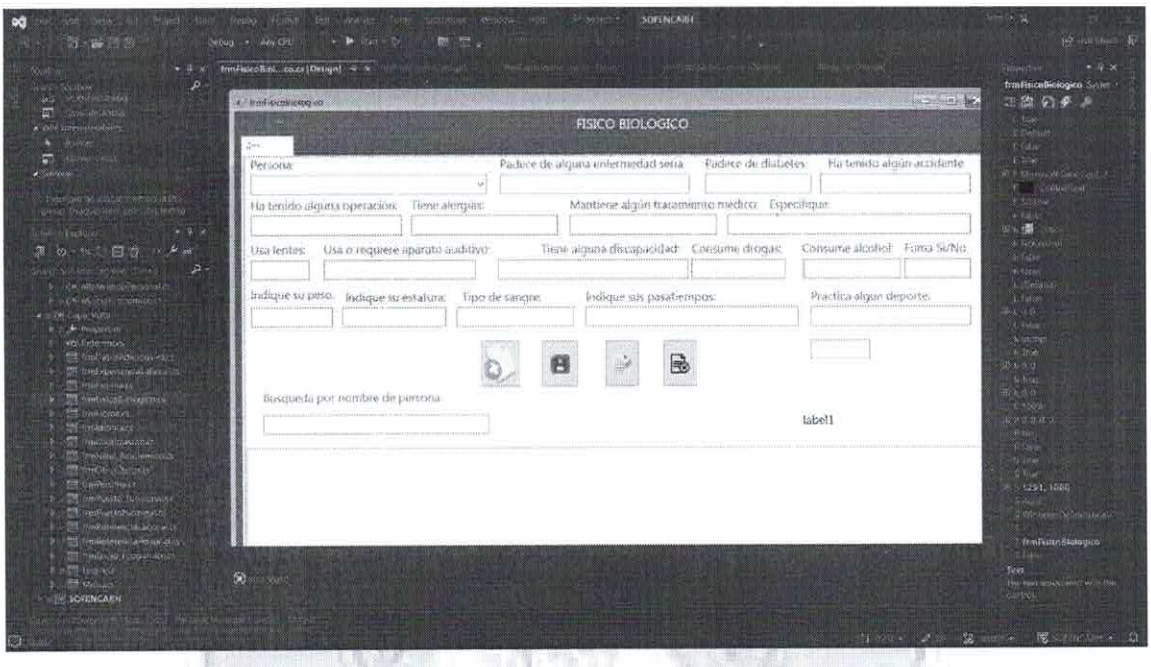
Aquí se ingresa la persona con todos sus datos que le corresponde.

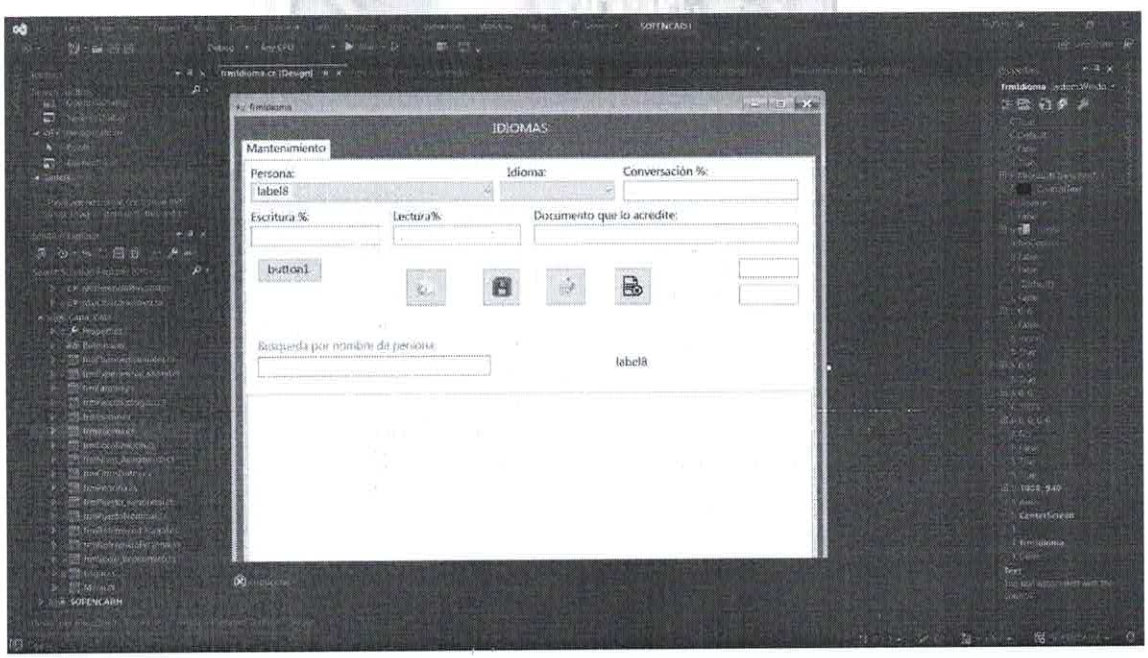
Ya teniendo la persona ingresada se debe de ingresar todos sus datos en las siguientes ventanas. Del sistema.

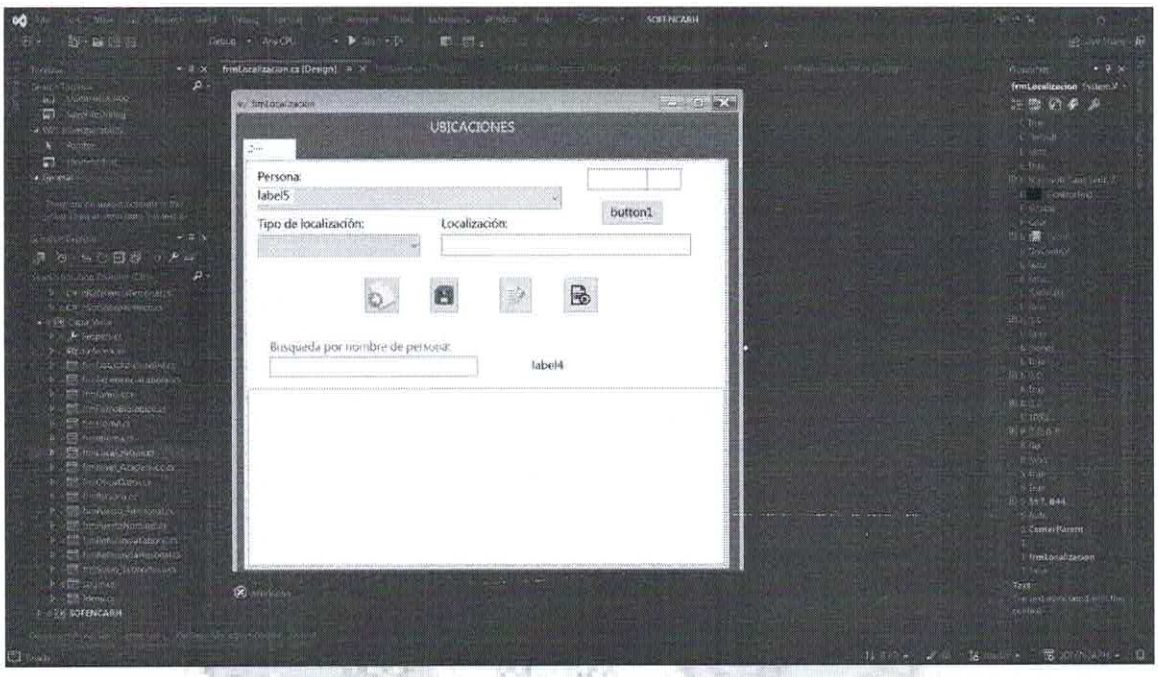


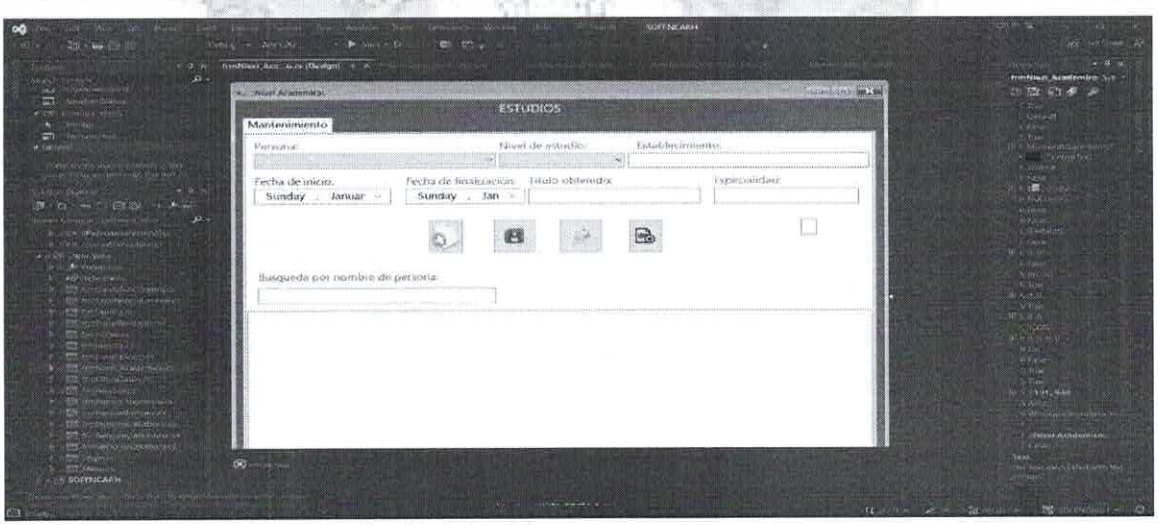

EXPERIENCIA LABORAL
Persona:
Empresa donde trabajo:
Telefono Empresa:
Jefe donde trabajo:
Puesto a cargo:
Salario devengado:
Fecha de Ingreso:
Fecha de retiro:
Motivo de retiro:
Se pueden solicitar referencias:
Descripcion de actividades que desempeñaba:
Busqueda por nombre de persona:
label1

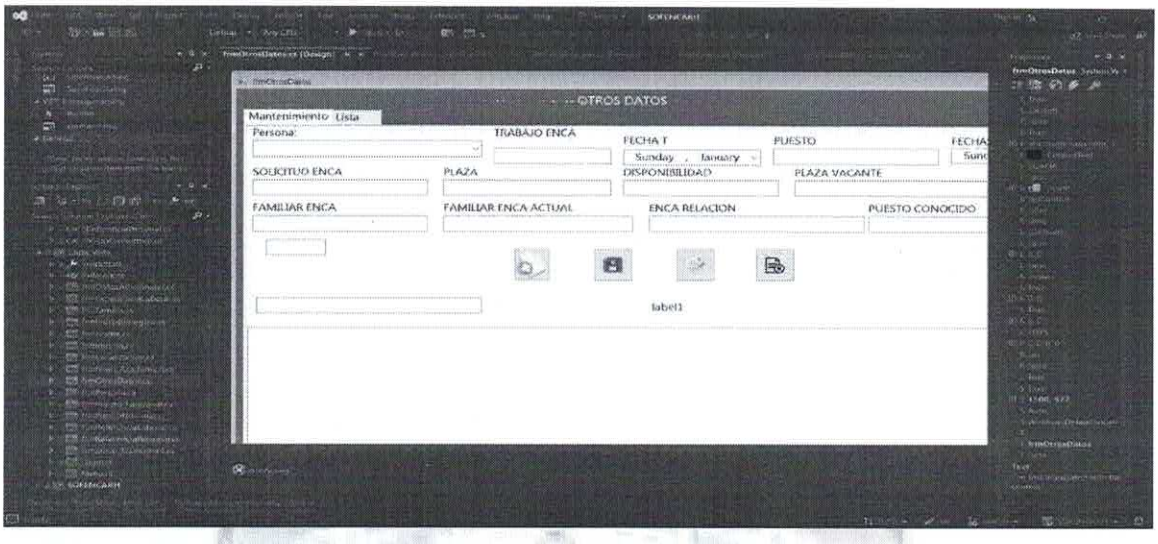

Familia.
Mantenimiento
Persona:
Tipo Familia:
Nombre familiar:
Fecha nacimiento familiar:
Edad:
Ocupación familiar:
Telefono de trabajo familiar:
Si su familiar Trabaja indique dirección y telefono de trabajo:
Lugar de trabajo familiar:
Dirección de trabajo familiar:
Telefono de trabajo familiar:
Busqueda por nombre de persona:
label10

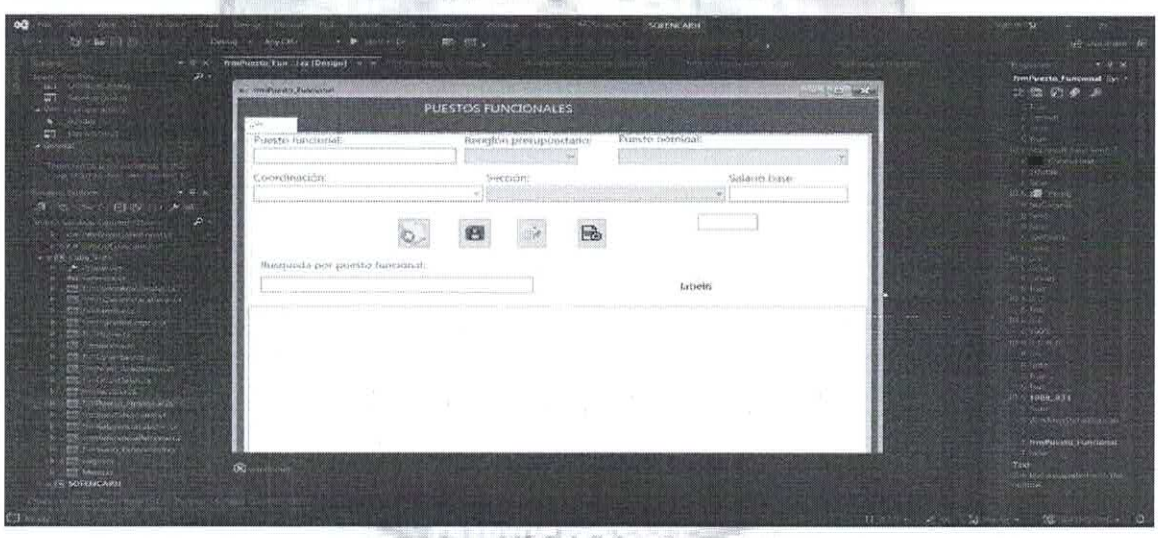

FISICO BIOLOGICO
Persona:
Padece de alguna enfermedad seria:
Padece de diabetes:
Ha tenido algún accidente
Ha tenido alguna operación:
Tiene alergias:
Mantiene algún tratamiento medico:
Especifique:
Usa lentes:
Usa o requiere aparato auditivo:
Tiene alguna discapacidad:
Consume drogas:
Consume alcohol:
Fuma Si/No
Indique su peso:
Indique su estatura:
Tipo de sangre:
Indique sus pasatiempos:
Practica algun deporte:
Busqueda por nombre de persona:
label1

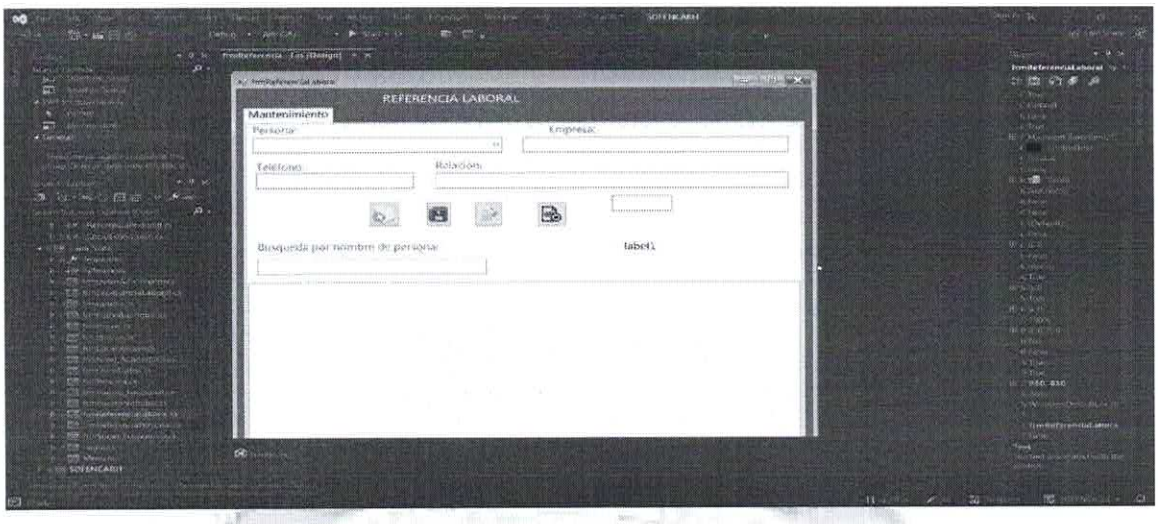

IDIOMAS
Mantenimiento
Persona:
label8
Idioma:
Conversación %:
Escritura %:
Lectura%:
Documento que lo acredite:
button1
Busqueda por nombre de persona:
label8

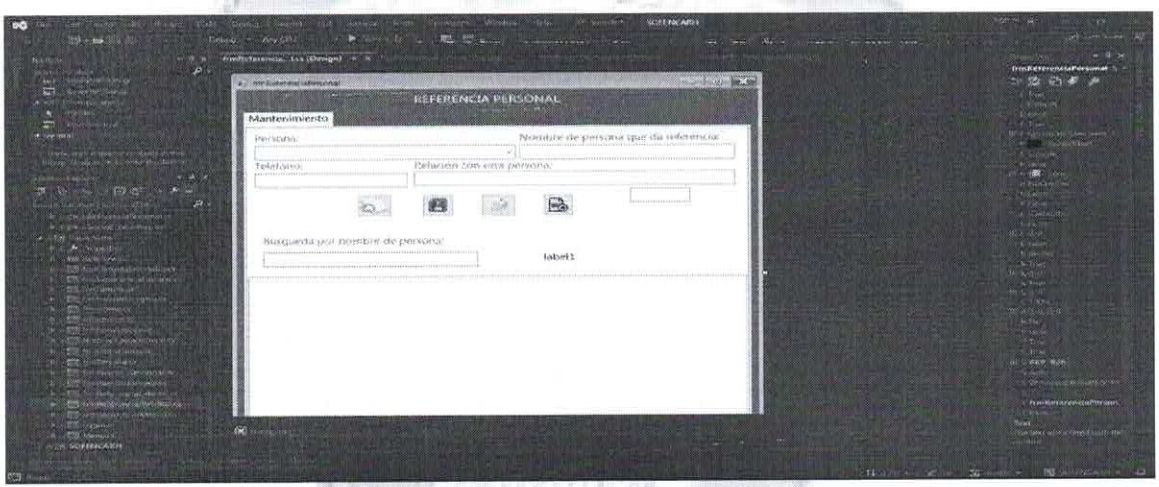

UBICACIONES
Persona:
label5
button1
Tipo de localización:
Localización:
Busqueda por nombre de persona:
label4

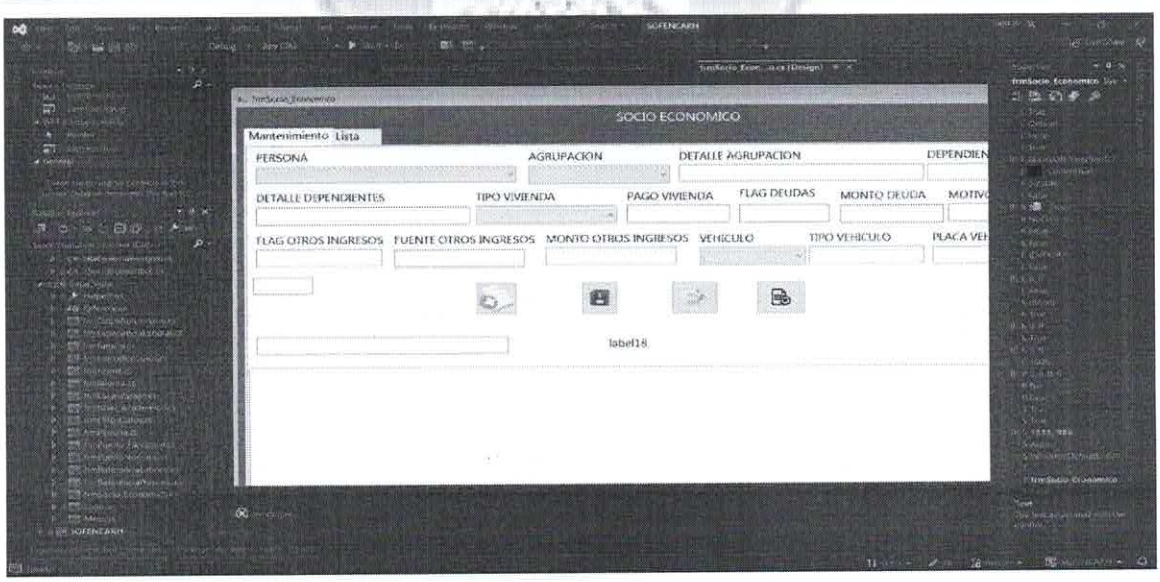

ESTUDIOS
Mantenimiento
Persona:
Nivel de estudio:
Establecimiento:
Fecha de inicio:
Sunday , Januar
Fecha de finalización:
Sunday , Jan
Título obtenido:
Especialidad:
Busqueda por nombre de persona:


OTROS DATOS
Mantenimiento
Lista
Persona:
TRABAJO ENCA
FECHA T
PUESTO
SOLICITUD ENCA
PLAZA
DISPONIBILIDAD
PLAZA VACANTE
FAMILIAR ENCA
FAMILIAR ENCA ACTUAL
ENCA RELACION
PUESTO CONOCIDO
label1


PUESTOS FUNCIONALES
Puesto funcional:
Región presupuestaria:
Puesto nominal:
Coordinación:
Sección:
Salario base:
Busqueda por puesto funcional:
label6

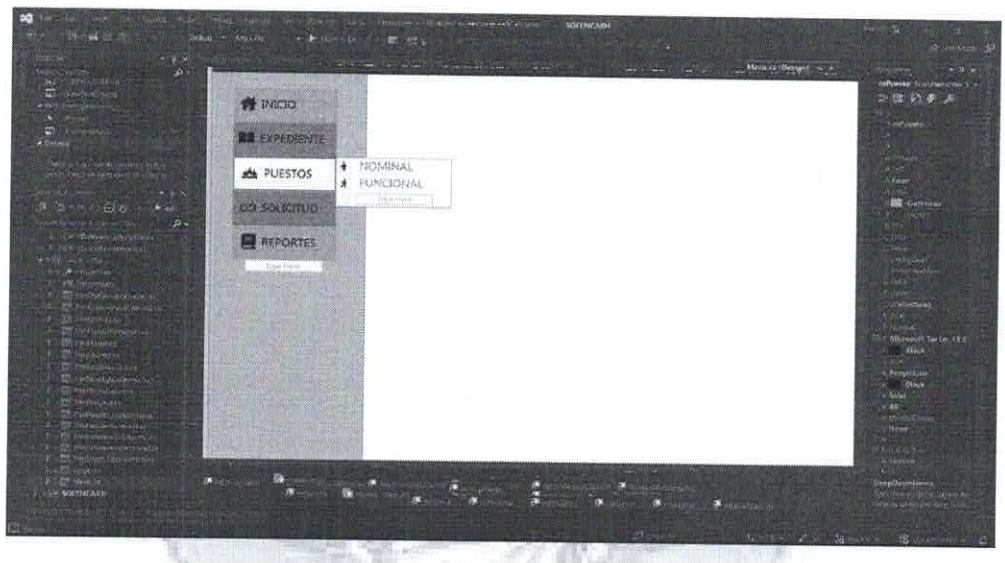

REFERENCIA LABORAL
Mantenimiento
Persona:
Empresa:
Telefono:
Relacion
Busqueda por nombre de persona:
label1

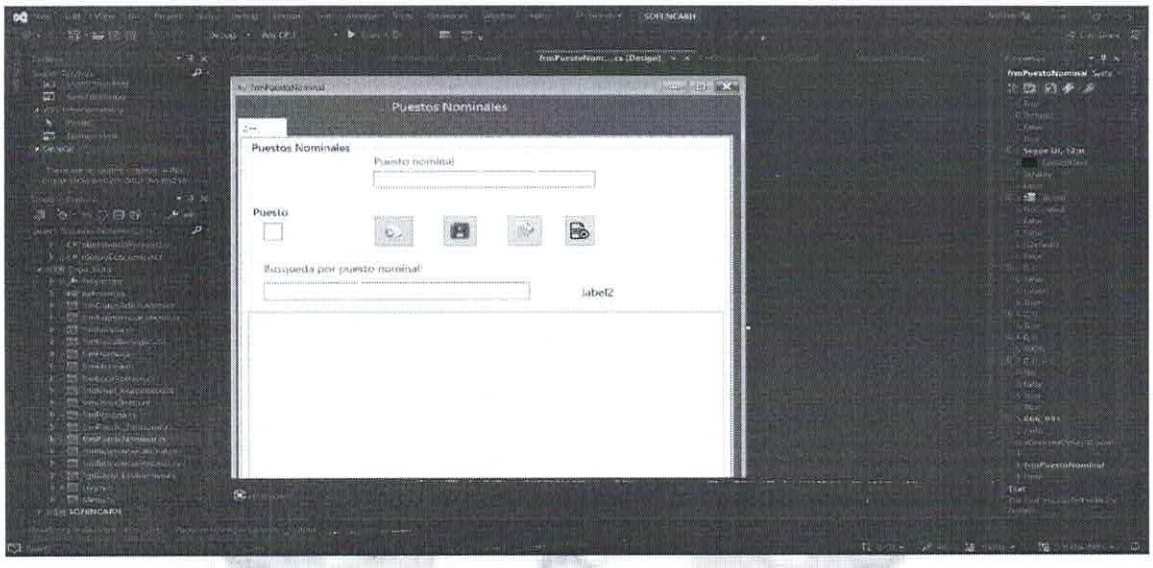

REFERENCIA PERSONAL
Mantenimiento
Persona:
Nombre de persona que da referencia:
Telefono:
Relacion con esta persona:
Busqueda por nombre de persona:
label1

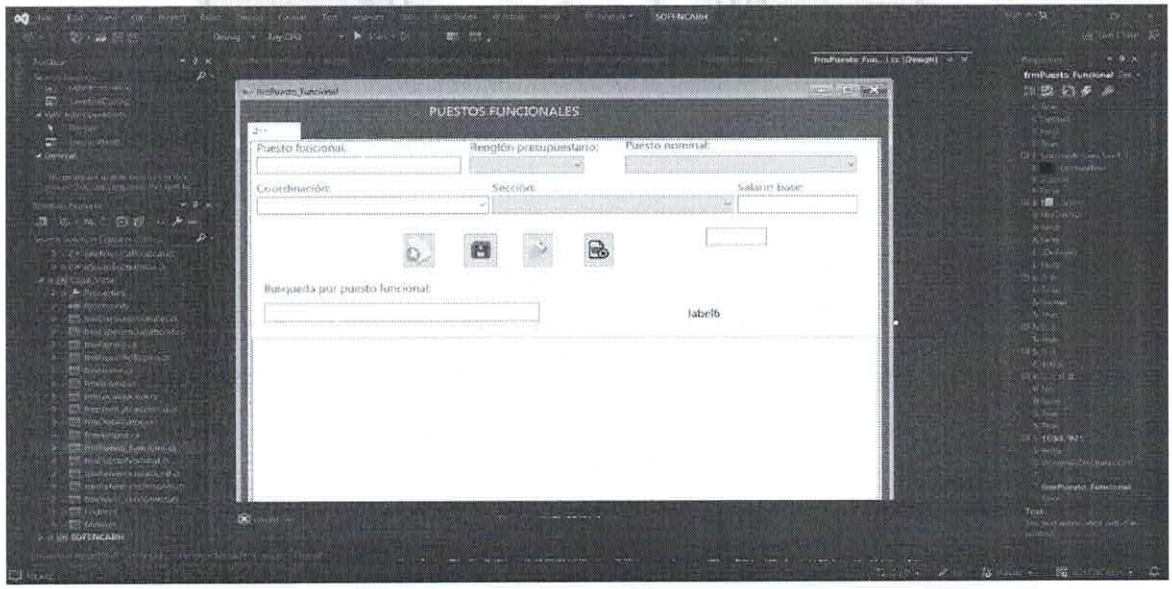

SOCIO ECONOMICO
Mantenimiento Lista
PERSONA
AGRUPACION
DETALLE AGRUPACION
DETALLE DEPENDIENTES
TIPO VIVIENDA
PAGO VIVIENDA
FLAG DEUDAS
MONTO DEUDA
FLAG OTROS INGRESOS
FUENTE OTROS INGRESOS
MONTO OTROS INGRESOS
VEHICULO
TIPO VEHICULO
label18

En términos generales estas son el diseño y lógica de la aplicación. Para hacer el ingreso de datos de la persona para crear su solicitud de empleo.

Luego regresando al menú. Aparece la parte de puestos aquí tendremos todos los puesto que se manejan la ENCA. Tanto puestos Nominales como puestos funcionales.





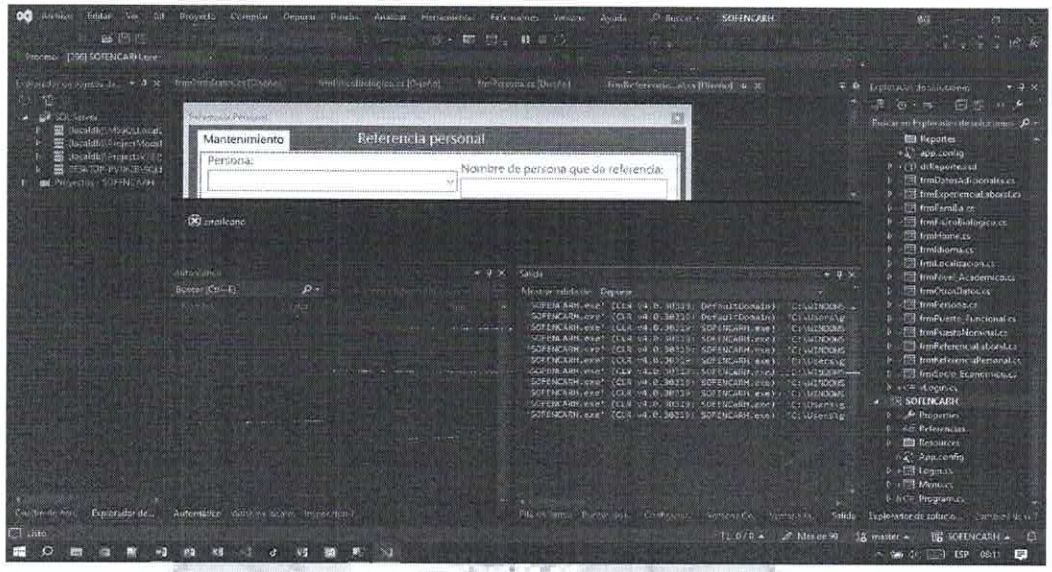


Luego aparece solicitudes y reportes esto se trabajará en este siguiente informe para dar por terminada lo que es el modulo de Admisión de personal y su toma de posesión del puesto.

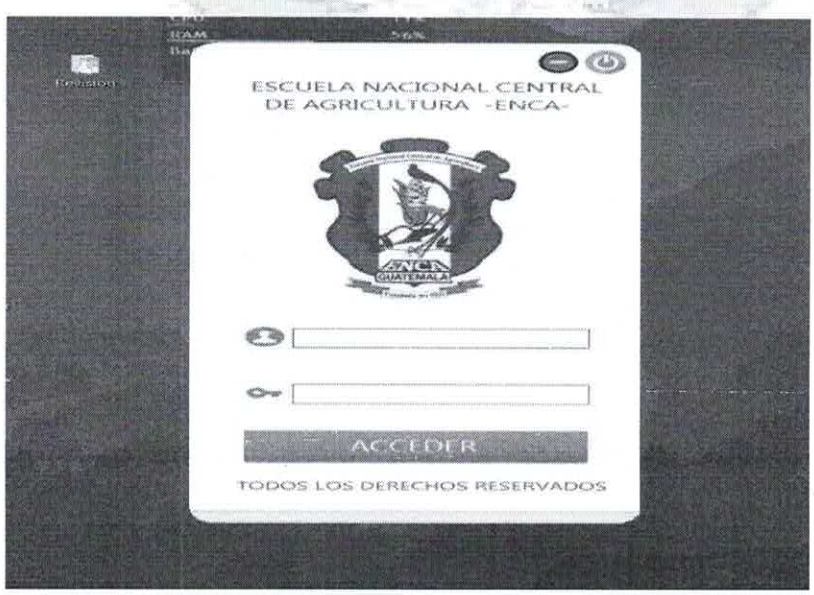
Compilación de Aplicación y funcionamiento de ingreso de solicitud completa de datos de una persona.

Paso 1.

Compilamos nuestra aplicación.



Y nos muestra nuestro login.

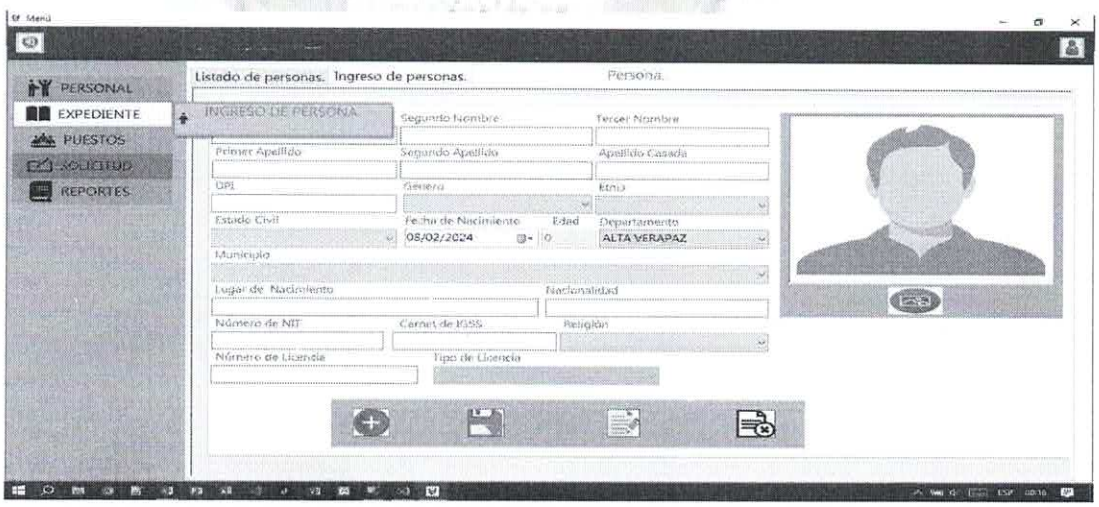


Aquí ingresamos nuestras credenciales.

Accedemos y nos mostrara el menú de nuestra aplicación.

En el inicio tenemos un menú que en la parte izquierda están las funcionalidades que se han desarrollado al sistema y en la parte derecha una imagen de la ENCA, como característica principal de la institución.

Para iniciar con la aplicación nos dirigimos a la opción donde dice expediente.

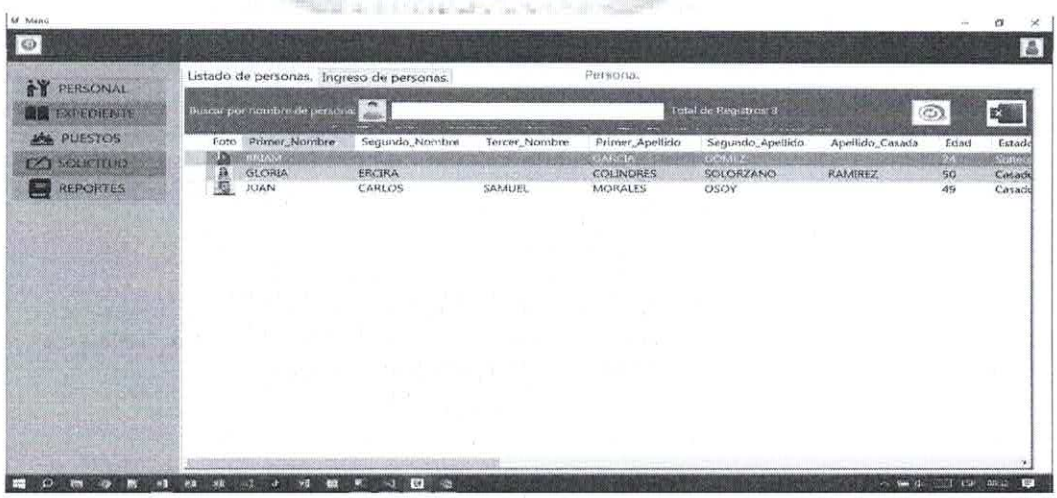


Y seleccionamos con un clic la opción donde dice ingreso de persona.

Luego ingresamos una persona donde tiene un panel con los botones que van hacer la función de nuevo registro, guardar, editar y cancelar la opción de ingreso de nueva persona.

Ingresamos datos de una persona para el ejemplo voy agregar los míos.

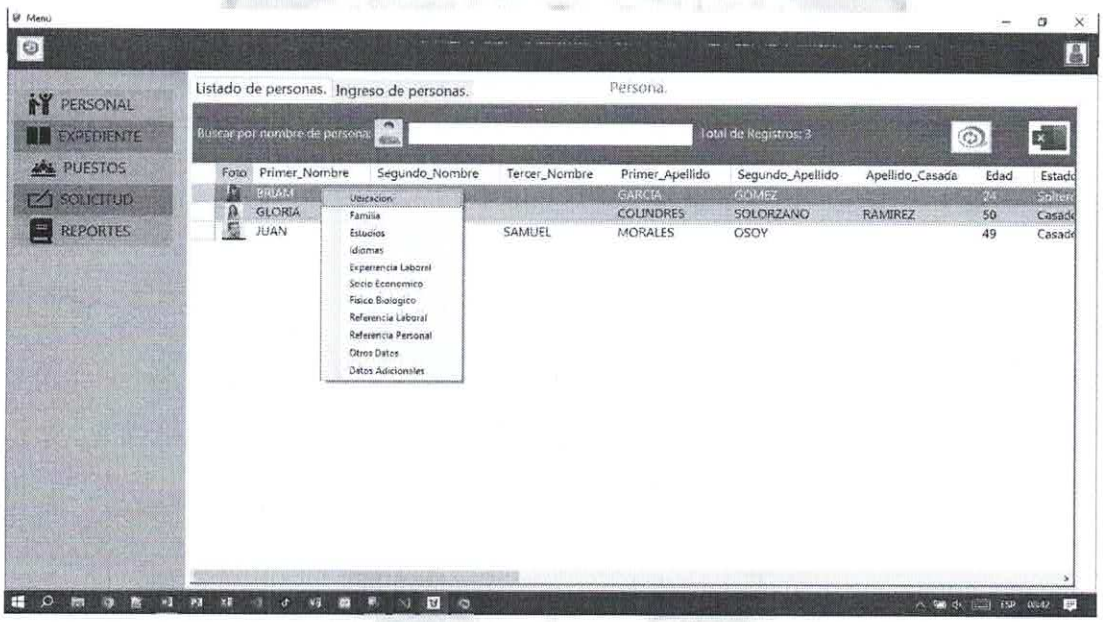
Ingresamos la persona y luego vamos a la pantalla de listado de persona donde nos muestra los datos de las personas que se han ingresado.



Ya al tener la persona ingresada podemos ingresar sus datos para llenar la solicitud de empleo.

Esta funcionalidad se maneja de la siguiente forma para poder insertar los datos de una forma personalizada por persona.

La lógica es la siguiente nos posicionamos sobre el registro de la persona, luego le damos clic derecho y nos despliega un sub menú donde esta las opciones de todas las informaciones que se le deben agregar a una persona en base a la solicitud.

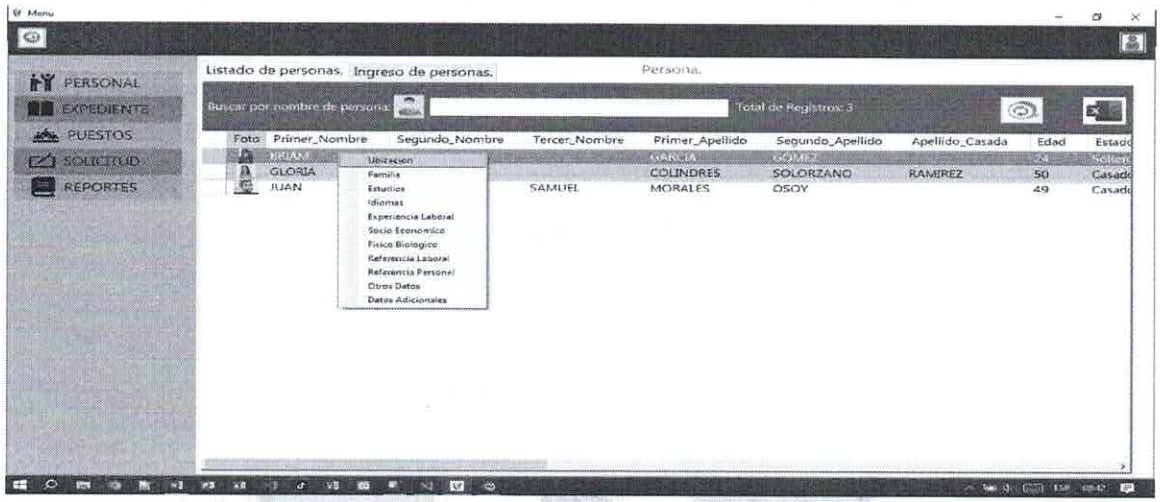


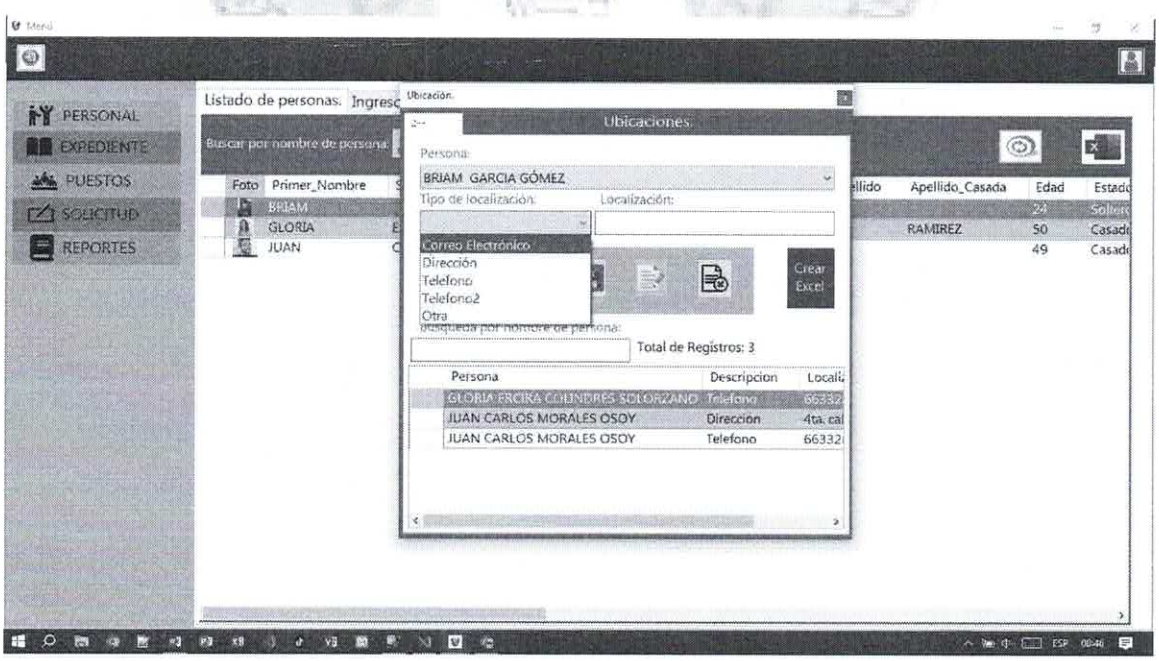
Como se puede ver el sub menú nos muestra los datos que se debe agregar a la persona para hacer una solicitud, esta funcionalidad esta para cada registro de persona es personalizada.

La lógica es la siguiente tenemos el submenú donde despliega opciones como ubicaciones, familia, idiomas entre otras.

Damos clic derecho y nos hace el llamado a la pantalla de la opción seleccionada.

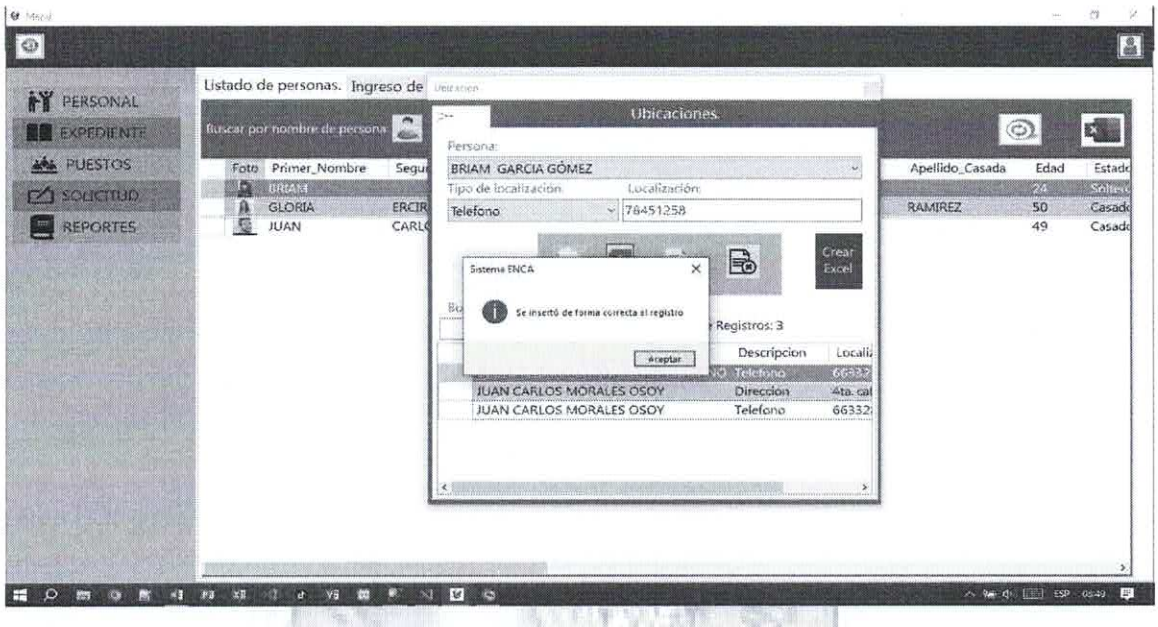
Veámoslo. Seleccionemos ubicación.





Nos muestra la ventana de ubicaciones aquí se le agregan ubicaciones a la persona, tenemos varias opciones de tipo de ubicaciones como podemos ver y en el

campo de Localización se agrega la ubicación ya sea un número de teléfono, una dirección o un correo electrónico.



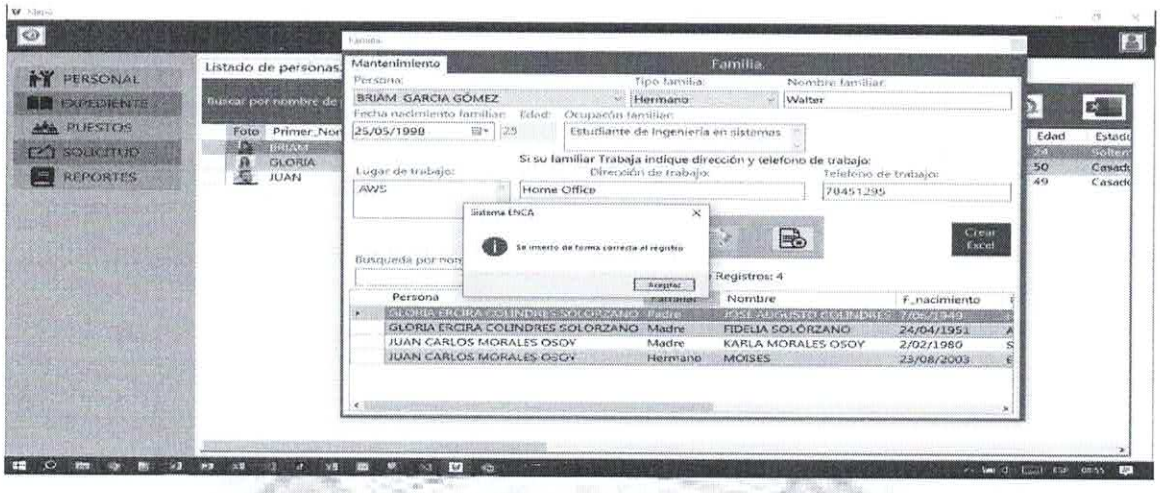
Aquí se le agrego un número de teléfono a Briam y nos dice un mensaje de validación que si se realizó el ingreso.

En términos generales la funcionalidad de ingreso de datos es así en todas las tablas, a que lleva esto a un sistema amigable con el usuario que sea fácil de utilizar.

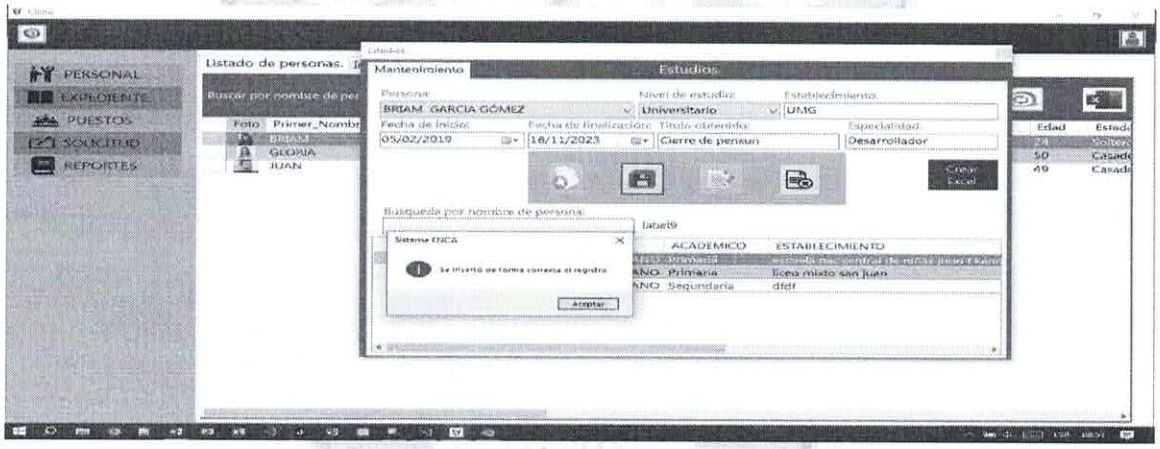
Y con la programación de capas que se está implementando para este sistema lo hace un sistema robusto y transaccionales a su base de datos.

A continuación, se hará de una forma breve el ingreso de todos los datos de la persona Briam para llenar la solicitud.

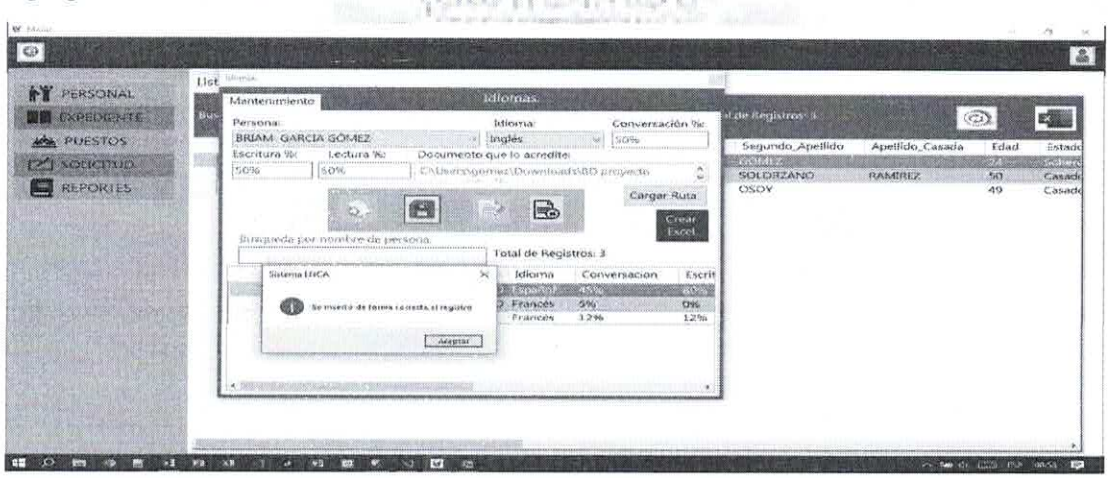
Agregamos un familiar.



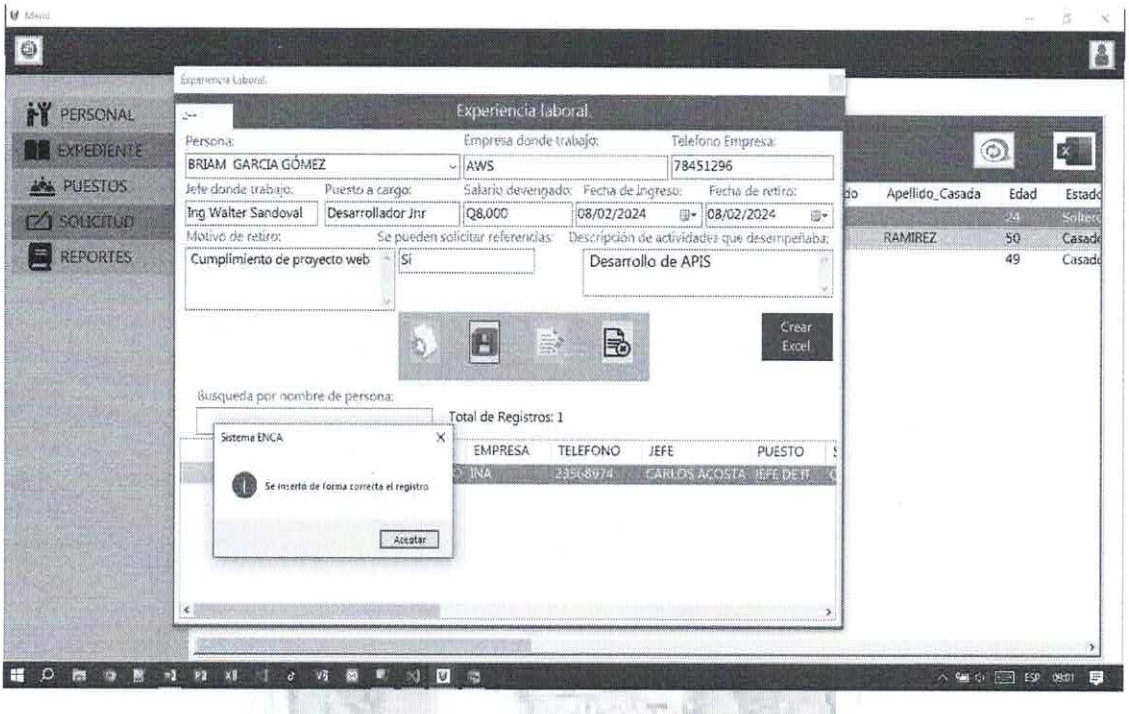
Agregamos un estudio.



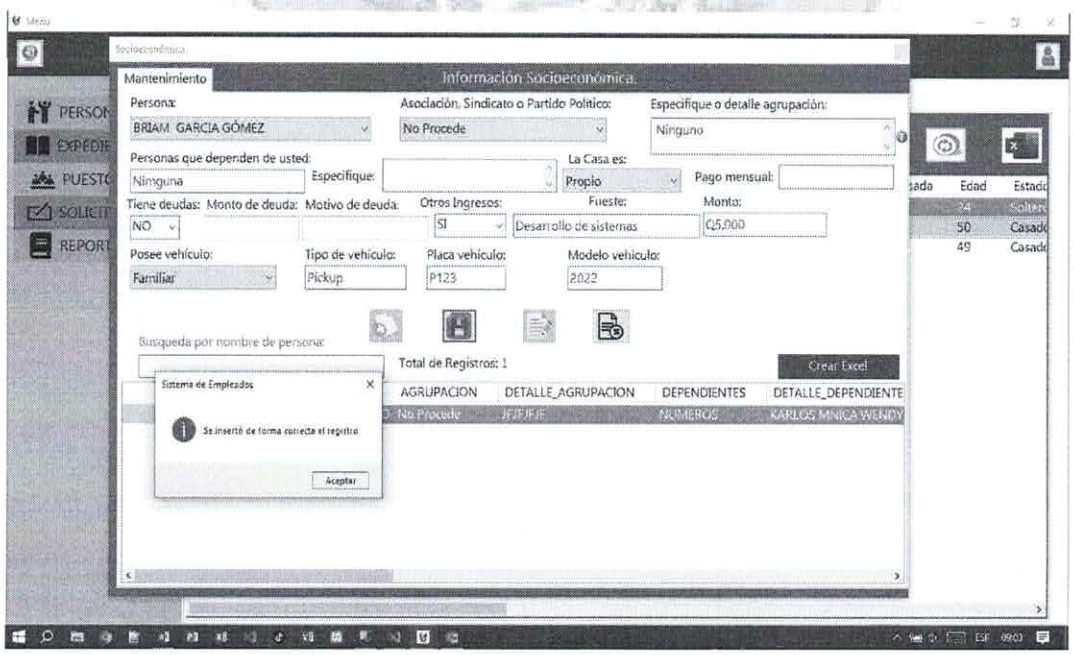
Agregamos un idioma.



Agregamos una experiencia laboral.

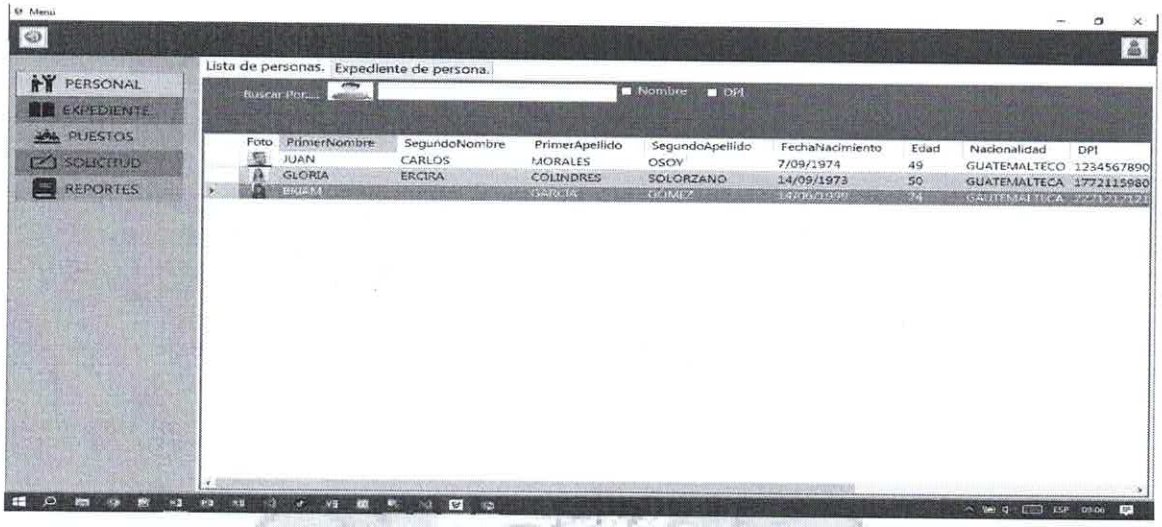


Agregamos información socioeconómica.



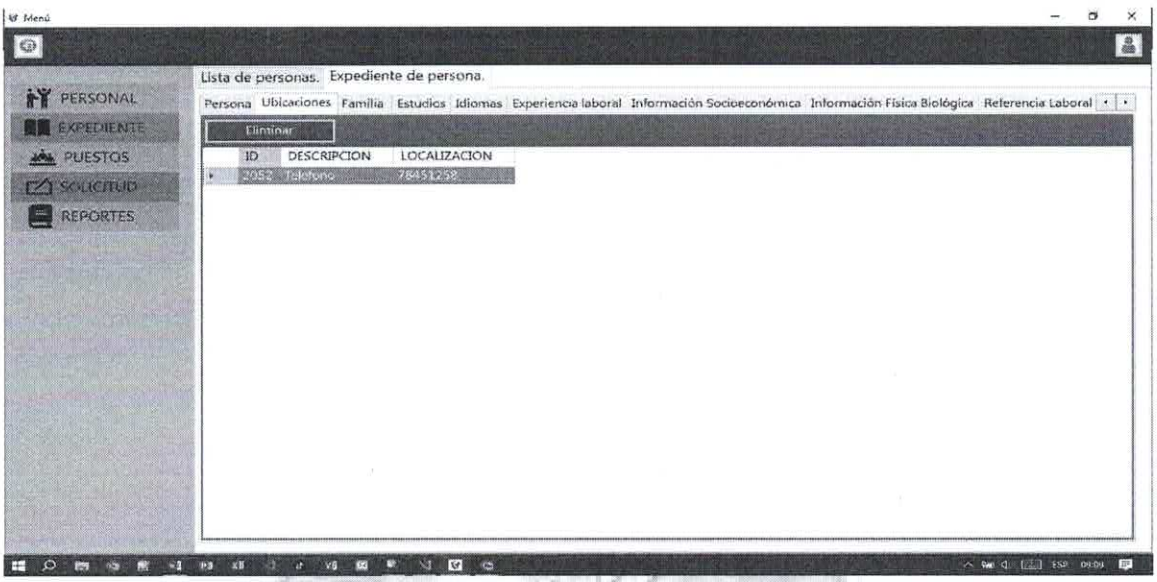
Y así nos vamos con todas las ventanas para el ingreso de la información de la persona.

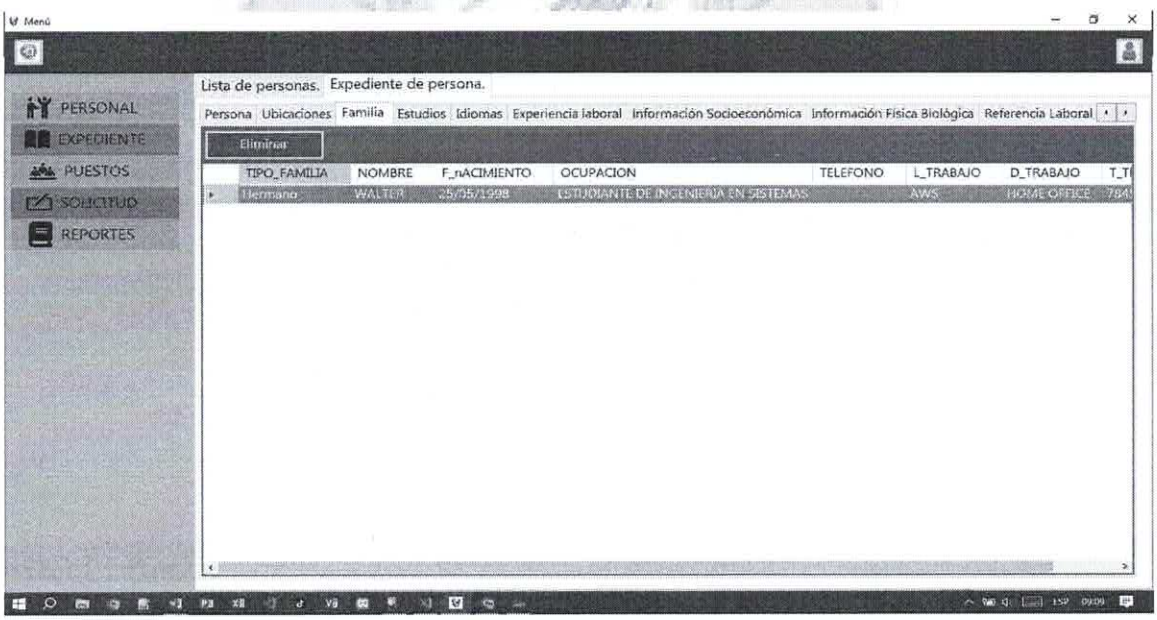
Ya teniendo el ingreso de todos los datos de la persona. Vamos al menú izquierdo de nuestra aplicación donde damos clic sobre la opción de personal.

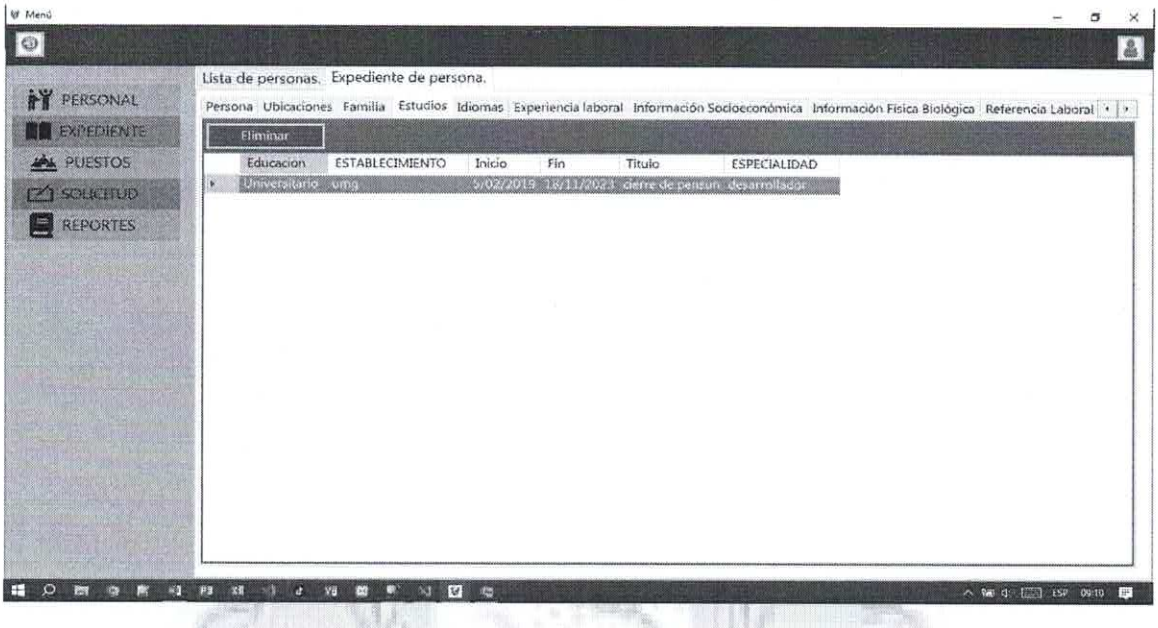


Aquí nos colocamos sobre la persona en el registro de datos de personal damos doble clic sobre el registro que deseamos, en mi caso voy hacerle sobre Briam.

Y en esta parte nos despliega datos de la persona y la imagen. A demás tiene un menú horizontal donde se va poder ver los datos de la persona que se ingresaron anteriormente. Veámoslos.

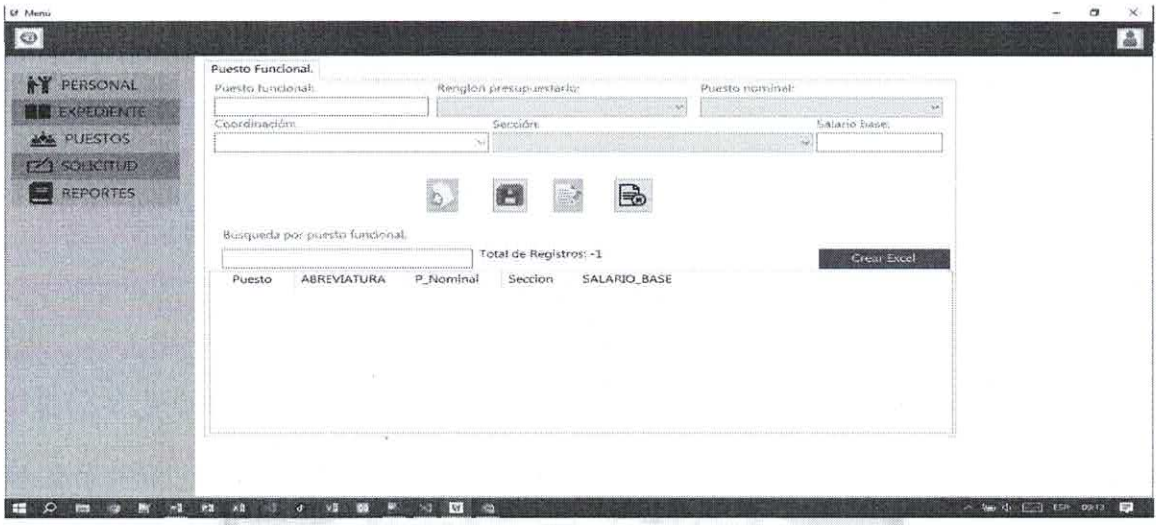






Y así en esta funcionalidad se selecciona a la persona y nos da una navegación donde tenemos los datos de la persona. Ya con este avance se tiene toda la solicitud de la persona donde se puede hacer un reporte con todos estos datos para poder imprimir esa información que esto se desarrollara en el siguiente avance de la aplicación.

También ya se tiene las demás pantallas. Como puestos nominales, puestos funcionales y renglones.



Todo esto para hacer reportes de las Actas entre otros documentos, con el puesto nominal, el puesto funcional y el renglón presupuestario.

Conclusión.

En el desarrollo de este proyecto, se crearon tablas mediante procedimientos cuidadosamente diseñados. La lógica de funcionamiento se implementó en C# utilizando una estructura de programación de capas para garantizar la modularidad y eficiencia del código. La aplicación se centra en el ingreso de datos de solicitud de empleo, proporcionando una interfaz intuitiva para el usuario. Posteriormente, se generan reportes detallados a partir de estos datos. Este enfoque estratégico asegura una gestión eficaz de la información, facilitando la toma de decisiones en el proceso de selección de personal.

Referencias.

https://visualstudio.microsoft.com/es/vs/community/

https://www.microsoft.com/es-es/sql-server/sql-server-downloads