

Proyecto Final de Ciclo

Dacobrick: ERP de apoyo a empresas constructoras.

Alumna: Sheila Soler Ramos
DNI: 73588369-F
Tutor Individual: Alicia González Canet
Tutor del grupo: Fidel Oltra Landete



Datos del Proyecto

Datos del Alumno

Nombre y apellidos	Sheila Soler Ramos
NIF/NIE	73588369-F
Curso y CF	2n DAM-SP

Datos del Proyecto

Título del proyecto	Dacobrick: ERP de apoyo a empresas constructoras.
Nombre del tutor individual	Alicia González Canet.
Nombre del tutor del grupo	Fidel Oltra Landete
Resumen	<i>Este proyecto consiste en diseñar, programar y desarrollar una aplicación de escritorio para el apoyo a empresas constructivas. Además, se crea y ajusta una base de datos según las funcionalidades planteadas. El objetivo es permitir el seguimiento de cada obra de forma completa. Dacobrick tiene en cuenta las múltiples necesidades que surgen en cuanto a obras, personal, documentación, etc.</i>
Abstract	<i>This project consists of designing, programming and developing a desktop application for the support of construction companies. In addition, a database is created and adjusted according to the proposed functionalities. The objective is to allow the monitoring of each construction site in a complete way. Dacobrick takes into account the multiple needs that arise in terms of construction sites, personnel, documentation, etc.</i>
Módulos implicados	<ul style="list-style-type: none"> • Bases de datos y acceso a datos. • Programación multimedia. • Desarrollo de interfaces. • Sistemas de gestión empresarial.
Fechas de presentación	14 de junio de 2021



Índice

Datos del Proyecto	2
Datos del Alumno	2
Datos del Proyecto	2
Índice	4
1. PRESENTACIÓN - JUSTIFICACIÓN	6
1.1. Descripción del proyecto	6
1.2. Objetivos	7
2. HERRAMIENTAS Y RECURSOS	8
3. CONTENIDO: DISEÑO Y DESARROLLO	9
3.1. Diseño: De la idea al proyecto	9
3.2. Desarrollo y configuración de la aplicación	11
3.3. Problemas encontrados y sus soluciones	57
4. AUTOEVALUACIÓN Y CONCLUSIONES	64
5. BIBLIOGRAFÍA	65
6. ANEXOS	67
6.1. Manual del usuario	67



1. PRESENTACIÓN - JUSTIFICACIÓN

1.1. Descripción del proyecto

El proyecto consiste en diseñar, programar y desarrollar una aplicación de escritorio para facilitar el trabajo a las empresas constructoras. A través de ésta se aúnan varias de las tareas diarias que se realizan en este tipo de empresas.

El ERP podría sumar distintas funcionalidades dependiendo del nivel profesional del usuario, es decir, en un futuro se podría adaptar a: Gerencia, trabajadores de obra, subcontratistas, etc. Por ejemplo, desde Gerencia se podría agregar la opción del control de vacaciones o un control general de ingresos y gastos; para los trabajadores de obra se podría incorporar un fichador o una pantalla para consultar horarios o indicaciones sobre los trabajos a realizar, etc.

En este caso, el prototipo desarrollado está centrado en los trabajadores técnicos, que es la parte en la que el ERP cobra mayor importancia, ya que éstos son los encargados de la organización diaria de una o varias obras.

Para la ejecución del proyecto se ha creado una base de datos con distintas tablas para guardar la información, esta tarea se ha realizado desde MySQL Workbench. Posteriormente, se han creado los distintos formularios desde Microsoft Visual Studio. Cada formulario presenta una pantalla distinta con información específica de obras, trabajadores, compras, etc.

Se han tenido en cuenta algunos programas similares como:

- Brickcontrol.
- m4PRO ERP.
- IB Building 365.

1.2. Objetivos

Actualmente, las necesidades que nacen de la dirección de las obras se cubren mediante distintas herramientas, documentación en papel, etc. Por ejemplo, las citas se organizan mediante una agenda clásica o gracias a funcionalidades como las que ofrece Google Calendar o algún sistema similar. Los contactos de la obra, tanto de trabajadores propios como de subcontratistas se guardan en la agenda del teléfono o en documentos escritos en papel. Otra de las tareas de los técnicos es llevar un seguimiento de los trabajadores en obra, lo más común es registrarlos mediante partes de trabajo en papel que se guardan en obra. También realizan otras funciones como compra de materiales, certificaciones de obra o presentación de proyectos a clientes potenciales. Todas estas funciones tienen un medio para llevárselas a cabo ya sea papel, en archivos de texto o de imagen, diferentes programas, etc.

El objetivo principal del ERP es unir estas funciones en una misma herramienta.

Otros objetivos son:

- Explotar un nicho de mercado, ya que su función se centra en un sector muy especializado como es la construcción.
- Mejorar la eficacia y la eficiencia empresarial, unificando las tareas y ofreciendo una plataforma sencilla e intuitiva.
- Compartir la información entre los trabajadores de la empresa, teniendo en cuenta sus funciones asignadas.
- Control de información personal como pueden ser horas y gastos.
- Eliminación de documentación en papel.

2. HERRAMIENTAS Y RECURSOS

Para el desarrollo del proyecto se ha utilizado:

- MySQL Workbench, una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL.
- Microsoft Visual Studio 2019, es un entorno de desarrollo integrado para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc. Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno compatible con la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y videoconsolas, entre otros.
- Github, repositorio online gratuito que permite gestionar proyectos y controlar versiones de código utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. Se podría hablar de Github como la red social pensada para desarrolladores, siendo este repositorio uno de los más usados a nivel mundial.
- C# o C Sharp, lenguaje de programación multiparadigma desarrollado y estandarizado por la empresa Microsoft como parte de su plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

3. CONTENIDO: DISEÑO Y DESARROLLO

3.1. Diseño: De la idea al proyecto

Para la presentación y el correcto funcionamiento de Dacobrick, el proyecto se compone de:

- Una base de datos realizada con MySQL Workbench. Esta base de datos integra distintas tablas para recoger la información generada:
 - Una tabla de obras con campos para registrar el identificador de la obra, su expediente y título, la dirección completa donde se desarrolla la obra. También se registra su estado, es decir, si está en estudio, en ejecución, finalizada, etc. Se guardan las fechas de inicio y finalización, los importes, el promotor, etc.
 - Una tabla para trabajadores con el código identificador de cada empleado, sus datos personales (nombre, apellidos, DNI y teléfono) y otros datos como si han firmado el contrato, si tienen una serie de cursos, si están asignados o no a una obra, etc.
 - Existen dos tablas para temas individuales como son el control de horas y el control de gastos. Para el control de horas se registran la hora de entrada, la hora de salida y su total por fecha asignándose a una obra concreta. Por su parte, en la tabla de gastos se registra el nº de factura, la fecha, el tipo (transporte, dieta, etc), el importe y también se fija a una obra en concreto.
 - Por último, existen una serie de tablas auxiliares para poder recoger datos respecto a documentación de las obras (antes del inicio, durante su desarrollo y al finalizarla), planificaciones, facturas, etc.
- La aplicación de escritorio realizada con Microsoft Visual Studio, compuesta por distintas pantallas desde las que se pueden ir realizando las tareas relacionadas con obras, personal, etc. La pantalla de inicio muestra el logo y un botón para acceder al menú. En el menú hay cuatro opciones: Obras, personal, planificación y otros.

Desde la opción de obras se pueden establecer nuevas obras con sus datos generales, planificaciones, documentación. Asimismo se pueden asignar operarios y registrar compras. Incluso se permite modificar las obras creadas por si existen variaciones en cuanto a su estado, importes, si se debe añadir/eliminar datos, asignar/eliminar trabajadores, etc.

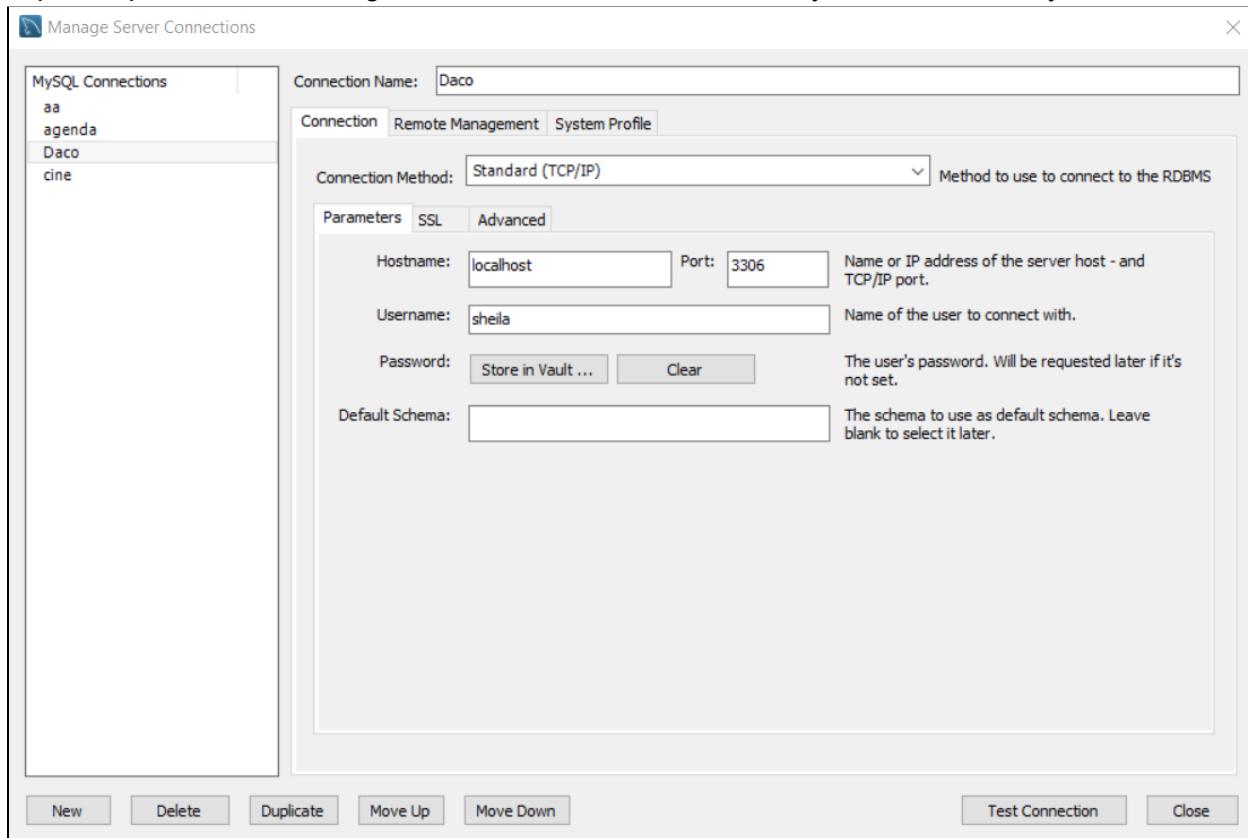
Desde la pestaña de personal se abre un submenú para, por una parte, registrar los gastos relacionados con transporte, dietas, alojamientos u otros. Por otra parte, guardar las horas dedicadas a cada obra. Además, se han creado dos gráficos de barras para que, visualmente, sea más sencillo extraer datos generales.

A través de la pestaña de planificación se accede a una pantalla en la que se puede seleccionar una obra o todas y una fecha. Pulsando el botón de mostrar se visualizan los eventos establecidos a la/s obra/s y la fecha indicada.

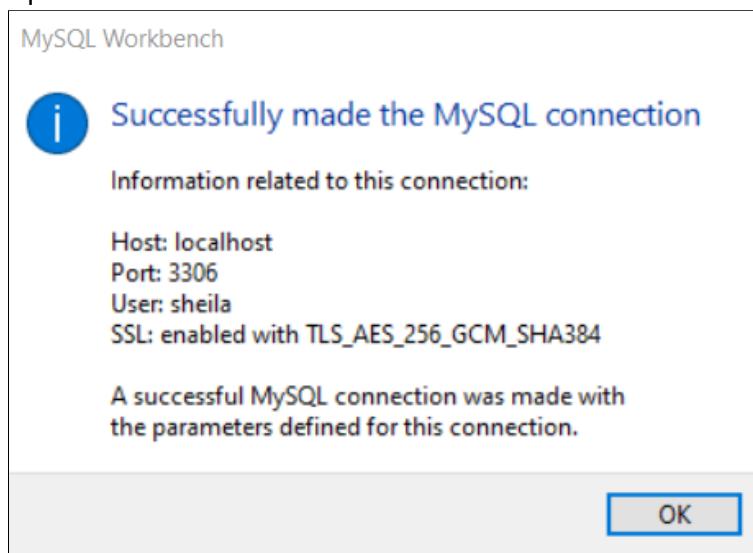
Para finalizar, desde otros, se accede a un submenú de trabajadores mediante el cual se pueden crear, modificar y/o eliminar operarios. En general se registran datos personales, el nivel profesional, la obra asignada, la documentación necesaria para empezar a trabajar (contrato, entrega de Equipos de Protección Individual, el apto médico, etc.), también los cursos realizados en cuanto a Prevención de Riesgos Laborales, albañilería, hormigón, etc.

3.2. Desarrollo y configuración de la aplicación

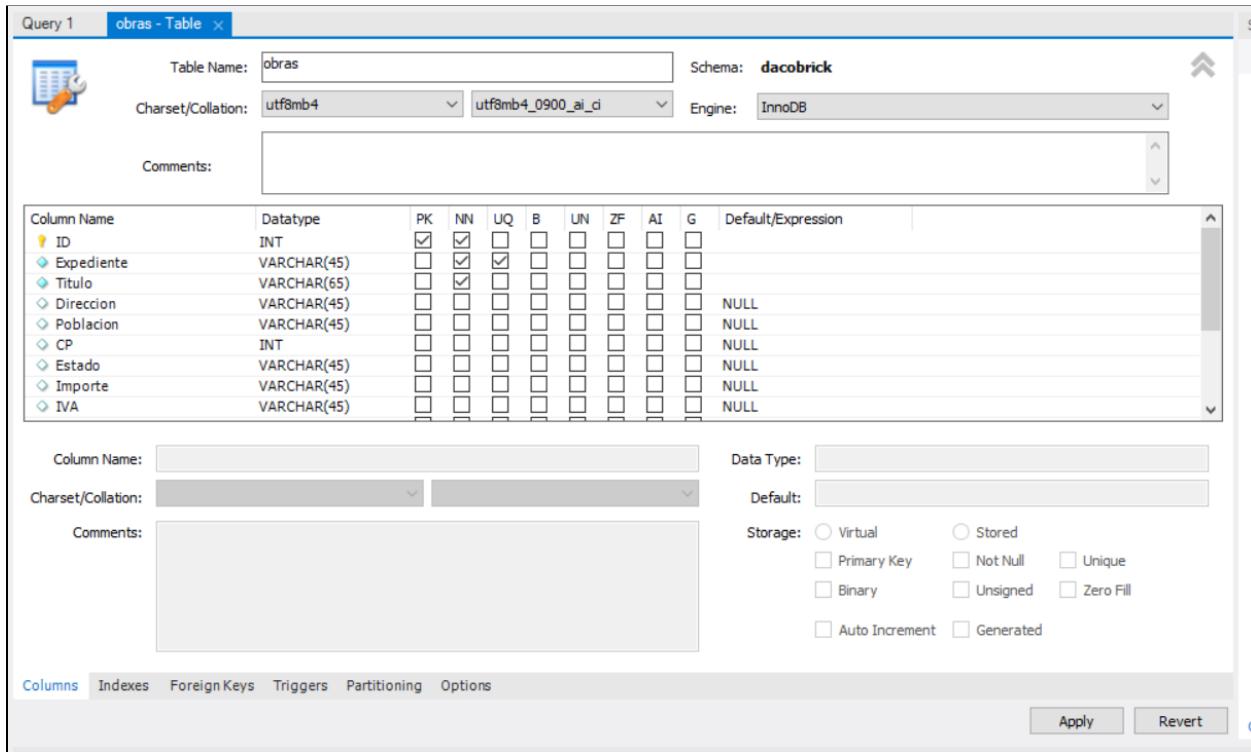
El primer paso ha sido configurar una nueva conexión entre MySQL Workbench y el servidor:



Se ha comprobado que la conexión es correcta:



A continuación, se ha creado una nueva base de datos con nombre “dacobrick” y las distintas tablas para guardar la información. Para crear las tablas, se ha introducido en cada una de ellas los campos necesarios con su nombre, el tipo de dato a introducir y, en su caso, la longitud máxima. También se ha indicado la clave primaria, si puede o no ser nulo el valor del contenido, si debe ser un valor único, o si debe ser o no autoincrementable, entre otros.



The screenshot shows the MySQL Workbench interface for creating a new table named 'obras'. The table is defined with the following columns:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>							
Expediente	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
Título	VARCHAR(65)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Dirección	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						
Población	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						
CP	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						
Estado	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						
Importe	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						
IVA	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						

Below the table definition, there are fields for adding a new column, specifying data type, default value, storage options (Virtual, Stored, Primary Key, Not Null, Unique, Binary, Unsigned, Zero Fill, Auto Increment, Generated), and buttons for Apply and Revert.

A continuación, se muestra el esquema final, la base de datos se compone de 9 tablas. La tabla principal es “obras”, ya que a partir de esta es de dónde aparecen el resto de tablas y sus relaciones.

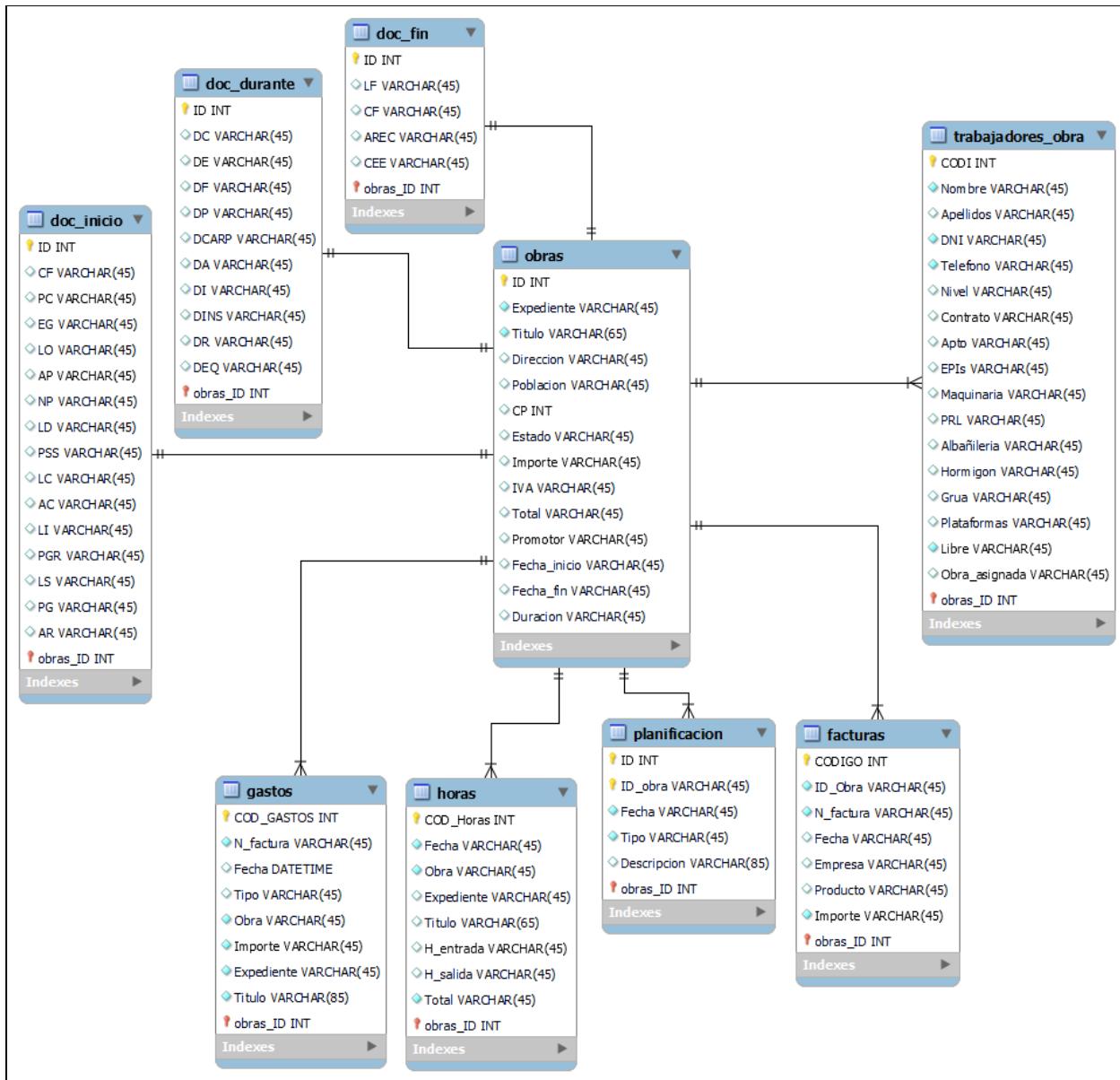
Como se puede observar en la imagen existe una relación 1:M entre la tabla “obras” y la tabla “trabajadores_obra”, ya que una obra puede tener muchos trabajadores, pero cada trabajador sólo podrá estar asignado a una obra.

Por otra parte, se encuentran las tablas “planificación” y “facturas” que también se relacionan con la principal con una relación 1:M, ya que se pueden establecer muchos eventos o realizar muchas facturas para una sola obra.

Las tablas “gastos” y “horas” son similares a las anteriores, pero se comentan aparte, ya que son tablas que, posteriormente, en la aplicación se sitúan en un submenú distinto. En este caso un gasto o un horario sólo pertenecerá a una obra, pero se pueden crear múltiples gastos y horarios para cada una de ellas.

Por último, las tablas de documentación, se han separado en tres bloques, dependiendo si los documentos son de inicio, se recogen durante la obra o se trata de documentación final de obra. Para estas tablas la relación es 1:1, ya que solamente existirá un tipo de documento para

cada obra, es decir sólo es necesario un contrato, un plan de seguridad y salud, un acta de recepción, etc.



El siguiente paso ha sido buscar un nombre que identifique al ERP, el nombre se compone de unas siglas (DACO) y la palabra BRICK. Por un lado, DACO significa Desarrollo de Aplicaciones para la Construcción y la Organización, con esto se pretende crear una palabra que pueda ser empleada en futuros productos y que muestre una imagen de conjunto. Se podría utilizar como DACOGestión, DACOMaquinaria, etc. dando a entender que forma parte de una serie de aplicaciones enfocadas al sector de la construcción. Por otro lado, la palabra BRICK significa ladrillo en inglés, es un guiño a que el ERP está relacionado con la edificación.

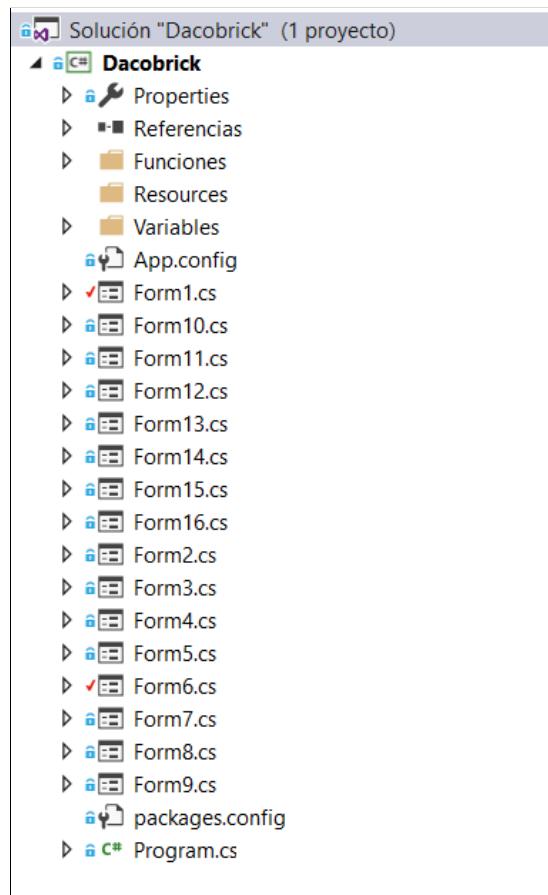
Asimismo se ha pensado en un eslogan: "¡Construye tu mundo!" y en un icono que se relacione fácilmente con ladrillos o un edificio. En cuanto, a los colores utilizados, se emplean los colores

blanco y negro para texto, fondos de pantalla, etc. También se utiliza el naranja, relacionándolo con el color de los ladrillos para los botones y pantallas.

El resultado final del nombre, icono, logo y colores se muestra en la pantalla de inicio:



El paso posterior ha sido realizar varios esquemas para crear una aplicación que sea intuitiva y fácil de manejar, aunque los datos a introducir sean variados, la intención es mostrar en cada pantalla exclusivamente la información necesaria. Antes de seguir con la explicación del proyecto, se muestra a continuación el esquema final de la aplicación en Microsoft Visual Studio:

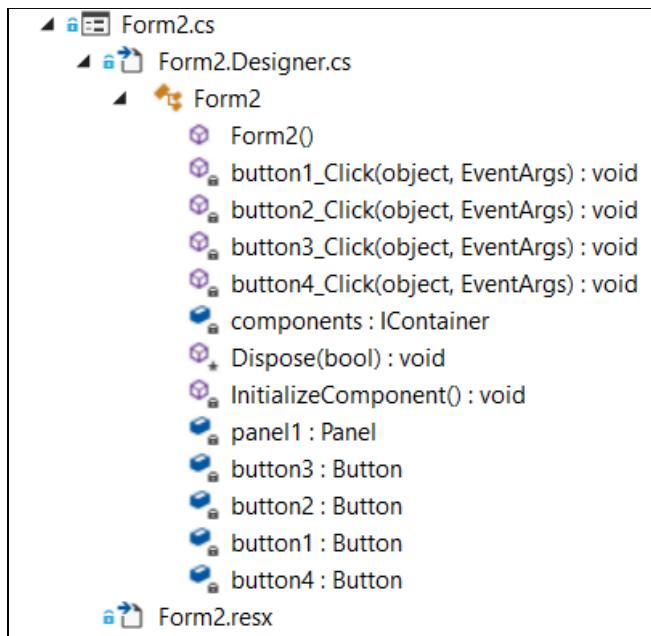


El proyecto “Dacobrick” se compone de un archivo “Main” cuyo contenido es:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Dacobrick
{
    static class Program
    {
        /// <summary>
        /// Punto de entrada principal para la aplicación.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Se han creado varias carpetas, una para funciones, otra para recursos y la última para variables. Además, se han ido creando los distintos formularios que se forman con un archivo gráfico y otro archivo de diseño.



Volviendo a la pantalla de inicio, sólo se ha incluido una acción que permite abrir el segundo formulario al hacer click en el botón “ENTRAR”:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Dacobrick
{
    public partial class Form1 : Form
    {

        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form frm = new Form2();
            frm.Show();
        }
    }
}
  
```

En cuanto al archivo de diseño del primer formulario, se puede observar que primero se crean los recursos a utilizar: Paneles, espacios de texto, botones, etc. y, seguidamente, se define para cada uno de ellos sus características como pueden ser: color de fondo, tamaño, posición, entre otras.

```

namespace Dacobrick
{
    partial class Form1
    {
        /// <summary>
        /// Variable del diseñador necesaria.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Limpiar los recursos que se estén usando.
        /// </summary>
        /// <param name="disposing">true si los recursos administrados se deben desechar; false en caso contrario.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Código generado por el Diseñador de Windows Forms

        /// <summary>
        /// Método necesario para admitir el Diseñador. No se puede modificar
        /// el contenido de este método con el editor de código.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Form1));
            this.panel1 = new System.Windows.Forms.Panel();
            this.button1 = new System.Windows.Forms.Button();
            this.label3 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label1 = new System.Windows.Forms.Label();
            this.panel1.SuspendLayout();
            this.SuspendLayout();

            // 
            // panel1
            // 
            this.panel1.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom | System.Windows.Forms.AnchorStyles.Left)));
            this.panel1.AnchorStyles.Right));
            this.panel1.BackColor = System.Drawing.Color.White;
            this.panel1.Controls.Add(this.button1);
            this.panel1.Controls.Add(this.label3);
            this.panel1.Controls.Add(this.label2);
            this.panel1.Controls.Add(this.label1);
            this.panel1.Location = new System.Drawing.Point(0, 0);
            this.panel1.Margin = new System.Windows.Forms.Padding(4);

```

```

this.panel1.Name = "panel1";
this.panel1.Size = new System.Drawing.Size(1340, 945);
this.panel1.TabIndex = 2;
//
// button1
//
this.button1.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Left) | System.Windows.Forms.AnchorStyles.Right));
this.button1.BackColor = System.Drawing.Color.Orange;
this.button1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.button1.Location = new System.Drawing.Point(467, 463);
this.button1.Margin = new System.Windows.Forms.Padding(4);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(420, 49);
this.button1.TabIndex = 3;
this.button1.Text = "ENTRAR";
this.button1.UseVisualStyleBackColor = false;
this.button1.Click += new System.EventHandler(this.button1_Click);
//
// label3
//
this.label3.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Left) | System.Windows.Forms.AnchorStyles.Right));
this.label3.Image = ((System.Drawing.Image)(resources.GetObject("label3.Image")));
this.label3.Location = new System.Drawing.Point(0, 0);
this.label3.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(1340, 412);
this.label3.TabIndex = 2;
this.label3.TextAlign = System.Drawing.ContentAlignment.TopCenter;
//
// label2
//
this.label2.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom | System.Windows.Forms.AnchorStyles.Left) | System.Windows.Forms.AnchorStyles.Right));
this.label2.BackColor = System.Drawing.Color.White;
this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 48F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.label2.ForeColor = System.Drawing.Color.Black;
this.label2.Location = new System.Drawing.Point(0, 764);
this.label2.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(1340, 134);
this.label2.TabIndex = 1;
this.label2.Text = "¡Construye tu mundo!";
this.label2.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;
//
// label1
//
this.label1.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom | System.Windows.Forms.AnchorStyles.Left) | System.Windows.Forms.AnchorStyles.Right));
this.label1.BackColor = System.Drawing.Color.White;

```

```
this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 72F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.label1.ForeColor = System.Drawing.Color.FromArgb(((int)((byte)(255))), ((int)((byte)(128))), ((int)((byte)(0))));  

this.label1.Location = new System.Drawing.Point(0, 635);
this.label1.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(1340, 162);
this.label1.TabIndex = 0;
this.label1.Text = "DACOBRICK";
this.label1.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;  

//  

// Form1  

//  

this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.AutoSize = true;
this.BackColor = System.Drawing.Color.White;
this.ClientSize = new System.Drawing.Size(1344, 897);
this.Controls.Add(this.panel1);
this.Margin = new System.Windows.Forms.Padding(4);
this.Name = "Form1";
this.Text = "DACOBRICK: Construye tu mundo.";
this.WindowState = System.Windows.Forms.FormWindowState.Maximized;
this.panel1.ResumeLayout(false);
this.ResumeLayout(false);  

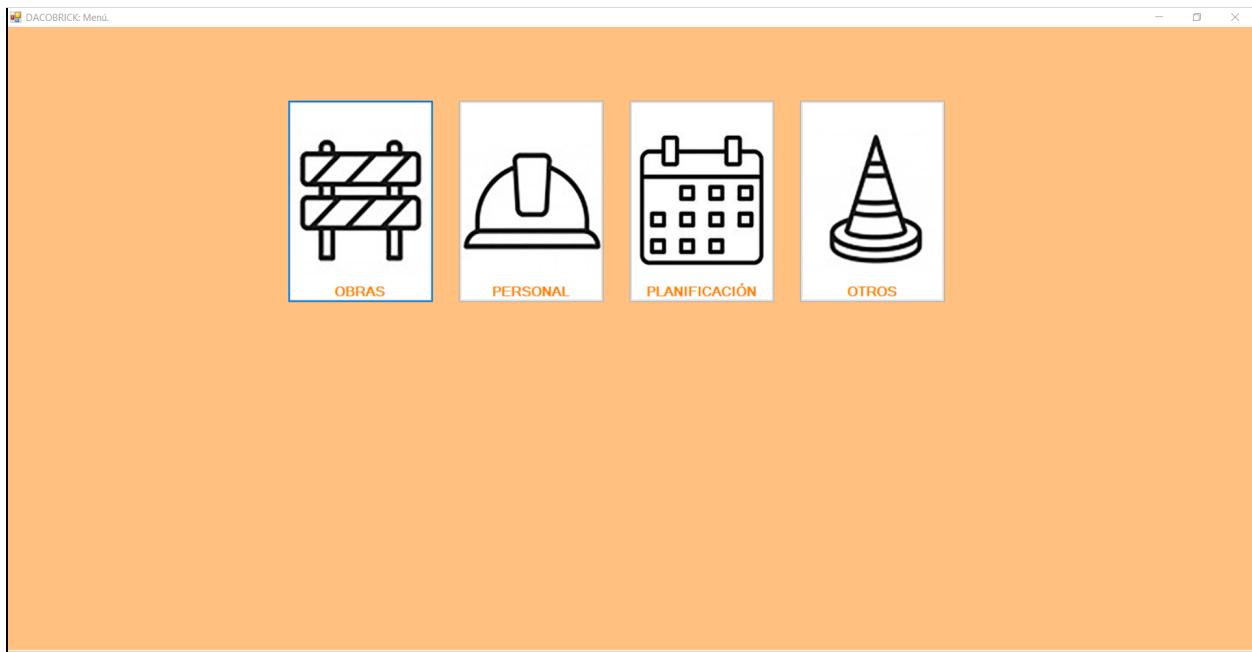
}  

#endregion  

private System.Windows.Forms.Panel panel1;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Button button1;  

}
```

El segundo formulario muestra el menú principal del ERP, se ha utilizado simbología propia de la construcción para los botones, además se ha incluido un texto debajo de cada uno para clarificar a qué parte de la aplicación se nos dirige al pulsarlo.



Para visualizar este formulario, primero se ha creado un panel dónde contener los botones, posteriormente se ha creado cada botón y se ha añadido un texto y una imagen. Al pulsar cada botón se comprueba si el siguiente formulario está abierto, en caso contrario se abre. El código empleado es el siguiente:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Dacobrick
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            bool Abierto = false;
            for (int i = 0; i < Application.OpenForms.Count; i++)
            {
                if (Application.OpenForms[i].Name == "Form3")
                {
                    Application.OpenForms["Form3"].Close();
                }
            }
            if (Abierto == false)
```

```

        {
            Form frm = new Form3();
            frm.WindowState = FormWindowState.Maximized;
            frm.Show();
        }
    }

private void button2_Click(object sender, EventArgs e)
{
    bool Abierto = false;
    for (int i = 0; i < Application.OpenForms.Count; i++)
    {
        if (Application.OpenForms[i].Name == "Form5")
        {
            Application.OpenForms["Form5"].Close();
        }
    }
    if (Abierto == false)
    {
        Form frm = new Form5();
        frm.Show();
    }
}

private void button3_Click(object sender, EventArgs e)
{
    bool Abierto = false;
    for (int i = 0; i < Application.OpenForms.Count; i++)
    {
        if (Application.OpenForms[i].Name == "Form8")
        {
            Application.OpenForms["Form8"].Close();
        }
    }
    if (Abierto == false)
    {
        Form frm = new Form8();
        frm.Show();
    }
}

private void button4_Click(object sender, EventArgs e)
{
    bool Abierto = false;
    for (int i = 0; i < Application.OpenForms.Count; i++)
    {
        if (Application.OpenForms[i].Name == "Form9")
        {
            Application.OpenForms["Form9"].Close();
        }
    }
    if (Abierto == false)
    {
        Form frm = new Form9();
        frm.Show();
    }
}
}

```

Desde la primera opción, “OBRAS” se accede a una pantalla con dos botones en la parte superior izquierda y el resumen de las obras ya creadas en forma de lista.



ID	Expediente	Título	Ubicación	Estado	Importe	Promotor	Fecha límite
1	348/2021bis	Adecuación del alumbrado público	Benifaió	PUBLICADA	265.240,01	Ayuntamiento de Benifaió	viernes, 27 de agosto de 2021
2	4289/2021	Resalfado de las avenidas de Almansa y Francisco Cerdá	Ontinyent	PUBLICADA	123.965,94	Ayuntamiento de Ontinyent	viernes, 27 de agosto de 2021
3	SIMPRAVACAMR.	Pavimentación de caminos rurales	Anna	EN EJECUCIÓN	65.774,18	Ayuntamiento de Anna	viernes, 27 de agosto de 2021
4	210/2021	Revestimientos interiores y urbanización (La Granaina)	Montavemar	ADJUDICADA A ...	45.454,55	Ayuntamiento de Montavemar	viernes, 27 de agosto de 2021
5	G-7619/2020	Reforma en Residencia 3º Edad "Virgen de Aguas Vivas"	Carcaixent	PRESENTADA	54.890,52	Ayuntamiento de Carcaixent	sábado, 26 de junio de 2021
6	2924/2021	Construcción de nichos y urbanización interior Cementerio	Algentí	EN EVALUACIÓN	239.362,63	Ayuntamiento de Algentí	viernes, 24 de septiembre de 2021
7	363/2021	Construcción canil bici en Alcàsser	Alcàsser	EN EJECUCIÓN	78.869,04	Ayuntamiento de Alcàsser	sábado, 31 de julio de 2021
8	77934BD	Mejora del Polígono Industrial Alquería de Raga	Picanya	EN EJECUCIÓN	114.882,26	Ayuntamiento de Picanya	jueves, 30 de septiembre de 2021
9	04101/2021/940	Reforma Exterior del Parque de Bomberos Oeste	Valencia	PUBLICADA	268.817,48	Ajuntament de València	martes, 31 de agosto de 2021
10	1012/2021	Ampliación escola infantil municipal "La Vila"	Albaida	FINALIZADA	1000000	Alcaldía del Ayuntamiento de Albaida	miércoles, 14 de julio de 2021

Se les ha añadido una acción “click” a cada botón cuyo código es el siguiente. Por una parte, el botón 1 representado con un signo “+” permite abrir el siguiente formulario, lo que hace es comprobar previamente si está abierto. El botón 2 con un icono de flecha permite cerrar el formulario actual y volver al anterior, en este caso, al menú principal.

```
private void button1_Click(object sender, EventArgs e)
{
    bool Abierto = false;
    for (int i = 0; i < Application.OpenForms.Count; i++)
    {
        if (Application.OpenForms[i].Name == "Form4")
        {
            Application.OpenForms["Form4"].Close();
        }
    }
    if (Abierto == false)
    {
        Form frm = new Form4();
        frm.ShowDialog();
        Cargar_Grid();
    }
}

private void button2_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Posteriormente se ha generado el código para cargar la lista, se trata de un elemento DataGridView y, para completar los datos se ha hecho una consulta a la base de datos. Como se puede observar, se hace uso de una función que se encuentra en el archivo “Conexiones”.

```
private void Cargar_Grid()
{
    DataSet ds = Conexiones.Retorna_Datos("SELECT * FROM obras ORDER BY ID");
    dataGridView1.AutoGenerateColumns = false;
```

```

        dataGridView1.DataSource = ds.Tables[0];
    }
}

```

A continuación se muestran las dos funciones que van a utilizarse a lo largo de todo el proyecto para realizar las conexiones entre el ERP y la BBDD. La primera función es para ejecutar las consultas, es decir, para realizar registros (INSERT), modificar/actualizar (UPDATE) los existentes o eliminar (DELETE) datos. La segunda función es para realizar consultas que devuelven información de la base de datos (SELECT).

En la primera, se utilizan las clases “OdbcConnection” y “OdbcCommand” para crear la conexión indicando la BBDD, en este caso “DACO”, el usuario “sheila” y la contraseña “toor”. Posteriormente, se comprueba si la conexión está cerrada o abierta, en el caso de estar cerrada se abre. Por último se pasa la consulta en SQL y se utiliza el método “ExecuteNonQuery” que permite instrucciones de INSERT, DELETE, UPDATE y SET .

En la segunda, se pasa también los datos para la conexión, es decir, nombre de la BBDD, usuario y contraseña. También se utiliza la clase “OdbcDataAdapter” para llenar un “DataSet” y actualizar los datos.

```

public static void Ejecuta_Consulta(string SQL)
{
    OdbcConnection cn = new OdbcConnection("dsn=DACO;uid=sheila;pwd=toor");

    try
    {
        OdbcCommand Cmd = new OdbcCommand(SQL, cn);

        if(cn.State == System.Data.ConnectionState.Closed)
        {
            cn.Open();
        }

        Cmd.CommandText = SQL;
        Cmd.ExecuteNonQuery();
        cn.Close();
    }
    catch(Exception ex)
    {
        cn.Close();
    }
}

public static DataSet Retorna_Datos(string SQL)
{
    OdbcConnection cn = new OdbcConnection("dsn=DACO;uid=sheila;pwd=toor");

    try
    {
        OdbcDataAdapter DA = new OdbcDataAdapter();
        DataSet ds = new DataSet();

        if (cn.State == System.Data.ConnectionState.Closed)
        {
            cn.Open();
        }

        DA.SelectCommand = new OdbcCommand(SQL, cn);
    }
}

```

```

        DA.Fill(ds, "0");
        cn.Close();

        return ds;
    }
    catch (Exception ex)
    {
        cn.Close();
        return null;
    }
}

```

Por último, tal y como se puede observar en la tabla de resumen de obras, existe al final de cada línea un botón con un icono de lápiz cuya función es editar los datos ya introducidos de la obra o añadir nuevas planificaciones, agregar documentación, etc. Esta función tiene el siguiente contenido, primero se crea una variable “NumFila” que será la que recoja dónde se ha ejecutado el click. Si cuando se pulsa sobre la gráfica se hace sobre la columna 8, se recogen los datos de identificador, expediente y título y se pasan a las variables globales. Posteriormente, se abrirá el formulario 4 que contiene la información más detallada de la obra seleccionada.

```

int NumFila = 0;
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    var senderGrid = (DataGridView)sender;

    try
    {
        if (senderGrid.Columns[e.ColumnIndex] is DataGridViewImageColumn && e.RowIndex >= 0)
        {
            NumFila = e.RowIndex;

            dataGridView1.CurrentCell = dataGridView1[0, NumFila];
            string ID_Eliminar = Convert.ToString(dataGridView1.Rows[NumFila].Cells[0].Value);

            //Editar
            if (e.ColumnIndex == 8)
            {
                string id = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
                string expediente = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
                string titulo = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();

                Variables_Globales.id = id;
                Variables_Globales.expediente = expediente;
                Variables_Globales.titulo = titulo;
                Variables_Globales.Form4_Desde = "FORM3";

                Form4 frm = new Form4();
                frm.ShowDialog();
                Cargar_Grid();

                Variables_Globales.Form4_Desde = "";
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error eliminando el registro indicado.");
    }
}

```

}

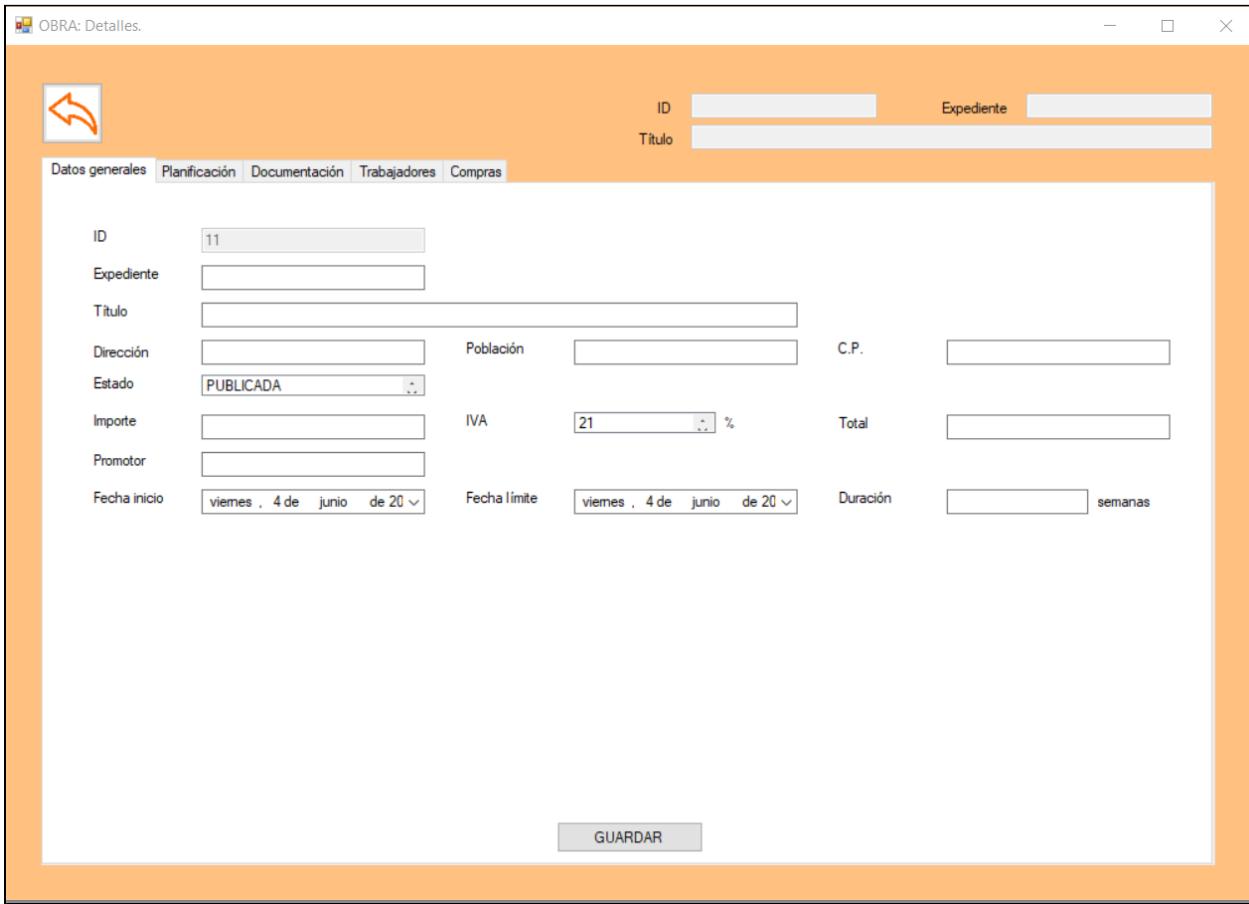
Ya que han aparecido anteriormente las variables globales y como se puede observar en el esquema de la aplicación, existe una carpeta de variables que contiene un archivo llamado “Variables_Globales”. En este se han definido una serie de elementos que van a utilizarse en varios formularios, su contenido se muestra a continuación:

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

namespace Dacobrick
{
    class Variables_Globales
    {
        public static string Identificador_obra = "";

        public static string id = "";
        public static string expediente = "";
        public static string titulo = "";
        public static string Form4_Desde = "";
        public static string codigo_empleado = "";
        public static string Form16_Desde = "";
    }
}
```

Tanto si se quiere modificar como crear una nueva obra, el formulario correspondiente es el 4. Está compuesto por: un botón de salir, cuyo funcionamiento ya se ha explicado anteriormente; también dispone de unos campos a la parte superior derecha que sirven de referencia para conocer en qué obra nos encontramos. Por último, contiene un elemento “tabControl” que permite incluir pestañas independientes en la misma ventana.



La primera función es la de cargar el formulario, en este caso, se diferencia si se trata de una nueva obra o se ha llegado a través del formulario 3, es decir, si la intención es modificar la obra existente. En cada caso, se ejecutan una funciones distintas. Si se trata de una obra nueva no se cargarán los datos porque todavía no existirán, ni las planificaciones, documentación, etc. porque todavía no se han creado.

```
private void Form4_Load(object sender, EventArgs e)
{
    if (Variables_Globales.Form4_Desde == "FORM3")
    {
        textBox15.Text = Variables_Globales.id;
        textBox16.Text = Variables_Globales.expediente;
        textBox17.Text = Variables_Globales.titulo;

        Cargar_Obra();
        Cargar_Grid_Planificacion();
        Cargar_Grid_Compras();
        Cargar_Inicio();
        Cargar_Durante();
        Cargar_Fin();
        Cargar_Grid_Trabajadores();
    }
    else
    {
        Buscar_ID();
    }
}
```

```

        Cargar_Grid_Planificacion();
        Cargar_Grid_Compras();
        Cargar_Grid_Trabajadores();
    }
}

```

En la primera pestaña se recogen los datos generales de la obra y, para guardarlos o modificarlos en la base de datos se debe pulsar el botón “GUARDAR”.

Se ha compuesto una función en la que el primer paso es crear las distintas variables para poder recogerlas desde su campo correspondiente (textBox, listBox o datePicker). A continuación, se comprueba si en la BBDD existe o no la obra, es decir, si el objetivo es modificarla o crearla. Si existe, se envía una orden de “UPDATE”, en caso contrario se envía una orden de “INSERT”. Por último, también se ha indicado que si alguno de los campos que no pueden ser nulos está vacío, aparezcan distintos mensajes informando al usuario que para continuar se deben completar dichos campos.

```

private void button2_Click(object sender, EventArgs e)
{
    string ID = "";
    string Expediente = "";
    string Titulo = "";
    string Direccion = "";
    string Poblacion = "";
    string CP;
    string Estado = "";
    string Importe = "";
    string IVA = "";
    string Total = "";
    string Promotor = "";
    string Fecha_inicio = "";
    string Fecha_fin = "";
    string Duracion = "";

    ID = textBox1.Text;
    Expediente = textBox2.Text;
    Titulo = textBox3.Text;
    Direccion = textBox4.Text;
    Poblacion = textBox5.Text;
    CP = textBox6.Text;
    Estado = listBox1.Text;
    Importe = textBox8.Text;
    IVA = listBox2.Text;
    Total = textBox10.Text;
    Promotor = textBox11.Text;
    Fecha_inicio = datePicker1.Text;
    Fecha_fin = datePicker2.Text;
    Duracion = textBox14.Text;

    if (textBox2.Text != "" && textBox3.Text != "" && listBox2.Text != "" && listBox1.Text != "")
    {
        string SQL = "Select * from obras where ID = " + ID + "";
        DataSet ds = Conexiones.Retorna_Datos(SQL);

        if (ds.Tables[0].Rows.Count > 0)
        {
            string SQL2 = "UPDATE obras set Estado = " + Estado + ", Importe = " + Importe + ", IVA = " + IVA +
", Total = " + Total + " , "

```

```

        "Fecha_inicio = '" + Fecha_inicio + "' , Fecha_fin = '" + Fecha_fin + "' , Duracion = '" + Duracion + "'"
        where ID = '" + ID + "'";

        Conexiones.Ejecuta_Consulta("UPDATE obras set Estado = '" + Estado + "' , Importe = '" + Importe + "' ,
        IVA = '" + IVA + "' , Total = '" + Total + "' , "
        "Fecha_inicio = '" + Fecha_inicio + "' , Fecha_fin = '" + Fecha_fin + "' , Duracion = '" + Duracion + "'"
        where ID = '" + ID + "'");

        textBox15.Text = ID;
        textBox16.Text = Expediente;
        textBox17.Text = Titulo;

        MessageBox.Show("Registro modificado correctamente.");

        tabControl1.SelectedIndex = 1;
    }
    else
    {
        string SQL2 = "INSERT INTO obras (ID, Expediente, Titulo, Direccion, Poblacion, CP, Estado, Importe,
        IVA, Total, Promotor, Fecha_inicio, Fecha_fin, Duracion) " +
            "VALUES ('" + ID + "' , '" + Expediente + "' , '" + Titulo + "' , '" + Direccion + "' , '" + Poblacion + "' , '" + CP
            + "' , '" + Estado + "' , '" + Importe + "' , '" + IVA + "' , '" + Total + "' , '" + Promotor + "' , '" + Fecha_inicio + "' , '" + Fecha_fin
            + "' , '" + Duracion + "')";

        Conexiones.Ejecuta_Consulta("INSERT INTO obras (ID, Expediente, Titulo, Direccion, Poblacion, CP,
        Estado, Importe, IVA, Total, Promotor, Fecha_inicio, Fecha_fin, Duracion) " +
            "VALUES ('" + ID + "' , '" + Expediente + "' , '" + Titulo + "' , '" + Direccion + "' , '" + Poblacion + "' , '" + CP
            + "' , '" + Estado + "' , '" + Importe + "' , '" + IVA + "' , '" + Total + "' , '" + Promotor + "' , '" + Fecha_inicio + "' , '" + Fecha_fin
            + "' , '" + Duracion + "')");

        textBox15.Text = ID;
        textBox16.Text = Expediente;
        textBox17.Text = Titulo;

        MessageBox.Show("Registro guardado correctamente.");

        tabControl1.SelectedIndex = 1;
    }
    else
    {
        if (textBox2.Text == "")
        {
            MessageBox.Show("Es obligatorio completar el EXPEDIENTE.");
        }

        if (textBox3.Text == "")
        {
            MessageBox.Show("Es obligatorio completar el TÍTULO.");
        }

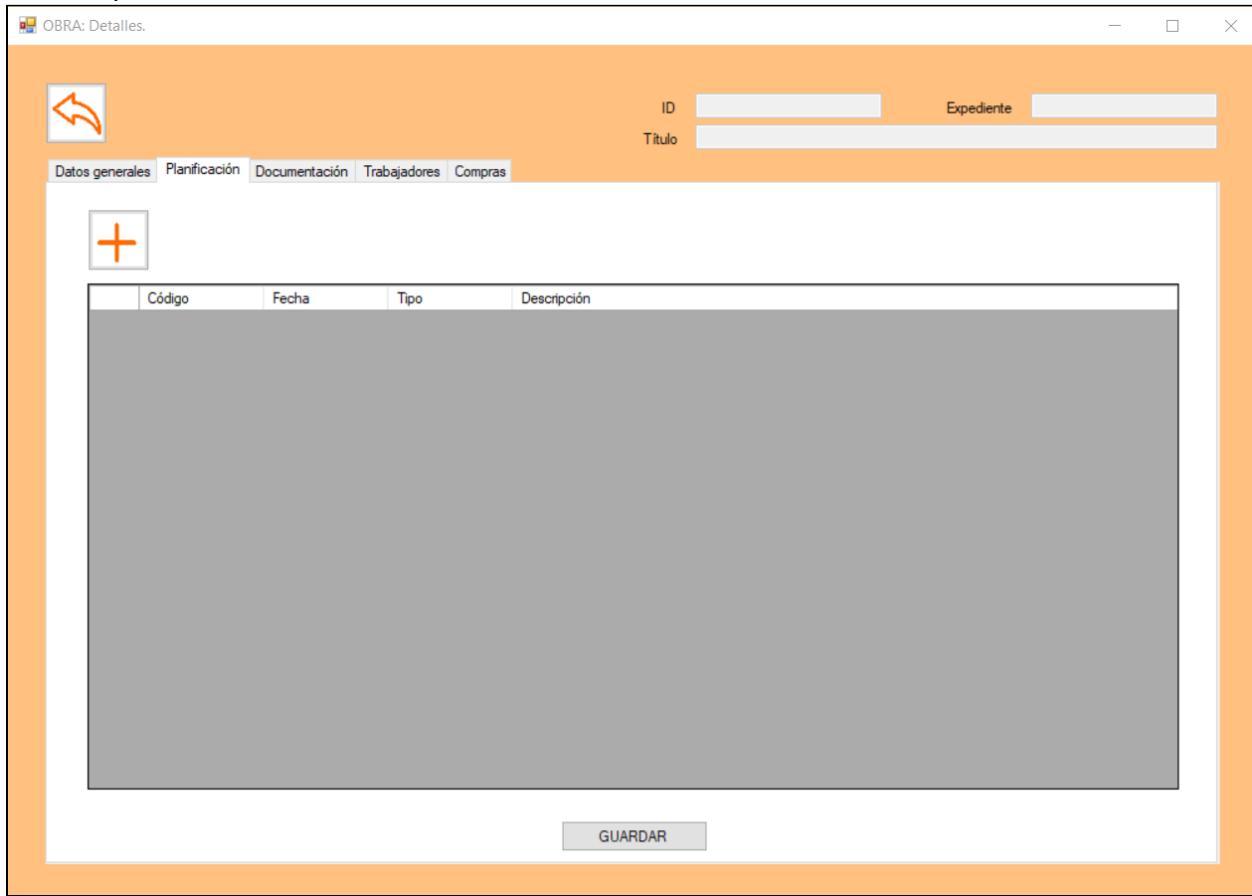
        if (listBox1.Text == "")
        {
            MessageBox.Show("Es obligatorio completar el ESTADO.");
        }

        if (listBox2.Text == "")
        {
            MessageBox.Show("Es obligatorio completar el IVA.");
        }
    }
}

```

}

La siguiente pestaña permite al usuario visualizar las planificaciones creadas para dicha obra a través de un “dataGridView” que carga la información realizando un “SELECT” a la BBDD del mismo modo que se hace en la tabla de resumen de obras. También permite crear una nueva planificación a través del botón “+”, cuya funcionalidad es abrir un nuevo formulario tal y como se ha explicado anteriormente.



Este nuevo formulario, muestra una serie de campos, el primero de ellos se rellena automáticamente. El resto deben ser completados por el usuario.

Nueva PLANIFICACIÓN

Código	13
Fecha	04-06-2021
Tipo	<input type="checkbox"/> Actuaciones previas <input type="checkbox"/> Demoliciones <input type="checkbox"/> Acondicionamiento del terreno <input type="checkbox"/> Cimentaciones <input type="checkbox"/> Estructuras
Descripción	
<input type="button" value="GUARDAR"/>	

El método que permite que el primer campo, de identificación de la planificación, se rellene automáticamente es el “Buscar_ID()”. Este método también se utiliza en otros formularios y el procedimiento es similar, se crea una variable que se utiliza en la función para devolver datos de la BBDD. Esta variable selecciona “SELECT” el máximo ID, es decir el valor mayor dentro de la columna de identificador de la tabla indicada, en este caso, “planificacion”. Si el máximo es igual a nulo es porque no existen todavía registros, por lo que el campo se llenará con un “1”, en el caso de que existan registros, recoge el valor máximo y le suma uno.

```
private void Buscar_ID()
{
    string maximo = "";
    try
    {
        String SQL = "SELECT MAX(ID) as ID FROM planificacion";
        DataSet ds = Conexiones.Retorna_Datos(SQL);
        maximo = Convert.ToString(ds.Tables[0].Rows[0]["ID"]);

        if (maximo == "")
        {
```

```

        textBox1.Text = "1";
    }

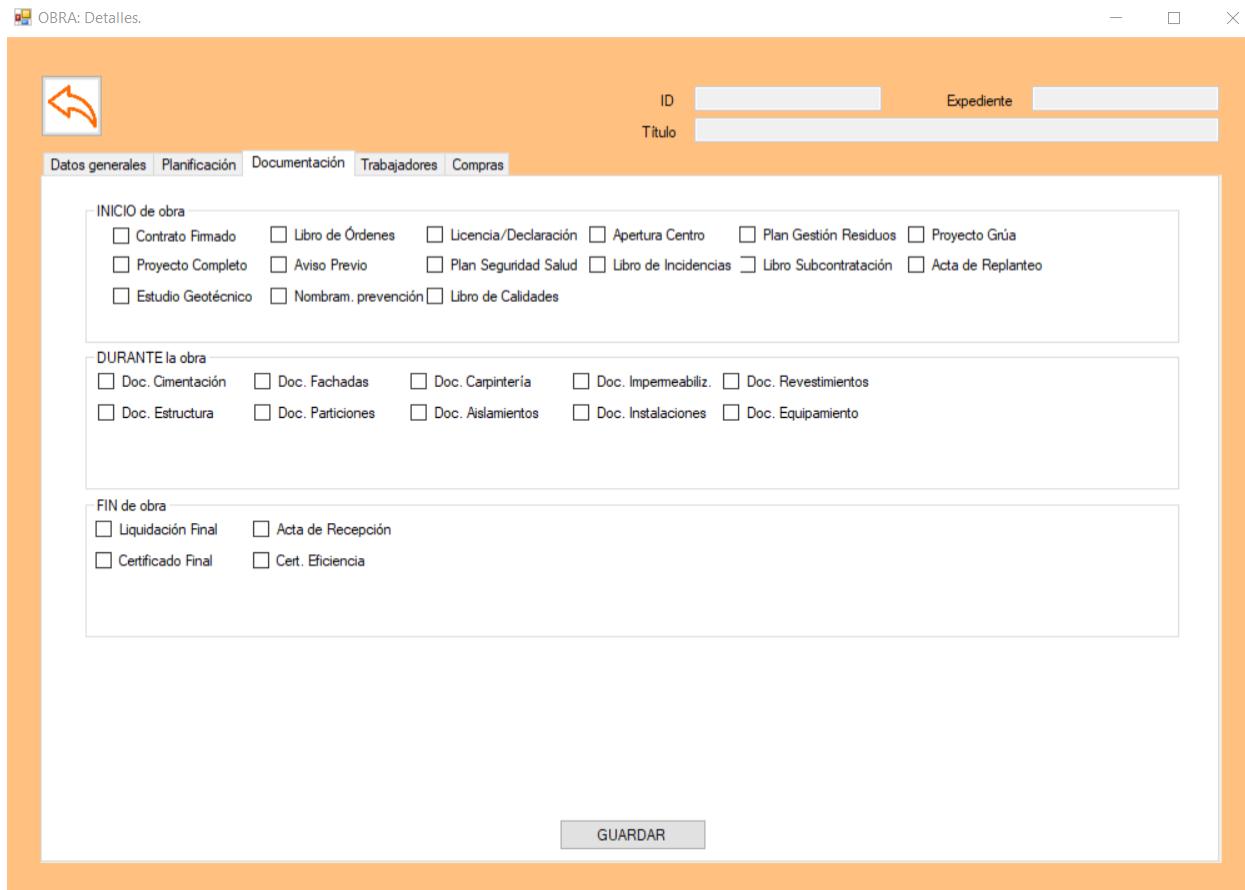
    if (maximo != "" && ds.Tables[0].Rows.Count > 0)
    {
        textBox1.Text = (Convert.ToInt32(ds.Tables[0].Rows[0]["ID"]) + 1).ToString();
    }
}

catch (Exception ex)
{
    //MessageBox.Show("Todavía no existían registros.");
}
}

```

La función de guardar tiene la misma estructura que la explicada con anterioridad, en este caso al no poder modificarse sólo se permiten realizar nuevos “INSERT” no “UPDATE”.

La siguiente pestaña es para anotar la documentación de la obra, por ejemplo, si se ha firmado el contrato se seleccionará la casilla correspondiente.



The screenshot shows a Windows application window titled "OBRA: Detalles". The main content area is an orange form with several sections:

- Header:** ID [text box], Expediente [text box], Título [text box].
- Buttons:** Back arrow icon, New icon, Save icon, Delete icon, Close icon.
- Navigation:** Datos generales, Planificación, Documentación (selected), Trabajadores, Compras.
- INICIO de obra:** A group of checkboxes for documents required at the start of the project, such as Contrato Firmado, Libro de Órdenes, Licencia/Declaración, etc.
- DURANTE la obra:** A group of checkboxes for documents required during construction, such as Doc. Cimentación, Doc. Fachadas, Doc. Carpintería, etc.
- FIN de obra:** A group of checkboxes for documents required at the end of the project, such as Liquidación Final, Acta de Recepción, Certificado Final, Cert. Eficiencia.
- Bottom right:** GUARDAR button.

En esta pantalla el botón guardar tiene una estructura un tanto diferente. Primero se han creado tres métodos, uno para guardar los documentos de cada bloque y a las tres se las llama desde el botón “GUARDAR”. Además, una vez se ejecutan los tres métodos, se indica que la

pestaña seleccionada debe ser la 3, de esta manera se consigue que una vez el usuario pulse en “GUARDAR” se cambie a la pantalla siguiente automáticamente.

```
private void button3_Click(object sender, EventArgs e)
{
    Guardar_Inicio();
    Guardar_Durante();
    Guardar_Final();
    tabControl1.SelectedIndex = 3;
}
```

El procedimiento que se emplea para guardar la documentación es similar en los tres casos, a continuación se comenta el “Guardar_Final()” por ser el más corto.

Primero, se recoge el valor del campo de texto que contiene el identificador de la obra, situado en la parte superior derecha y se ejecuta un “SELECT” a la BBDD para comprobar qué documentación está ya seleccionada. Sólo dispondrá de documentos seleccionados si es una modificación, en una obra nueva no existirá documentación. A continuación, se crean las variables y se guardan los valores de los distintos checkBox.

Si la consulta retorna datos, es decir “ds.Tables[0].Rows.Count > 0” se realiza un “UPDATE”, en caso contrario se ejecuta un “INSERT INTO”.

```
private void Guardar_Final()
{
    Variables_Globales.id = textBox15.Text;

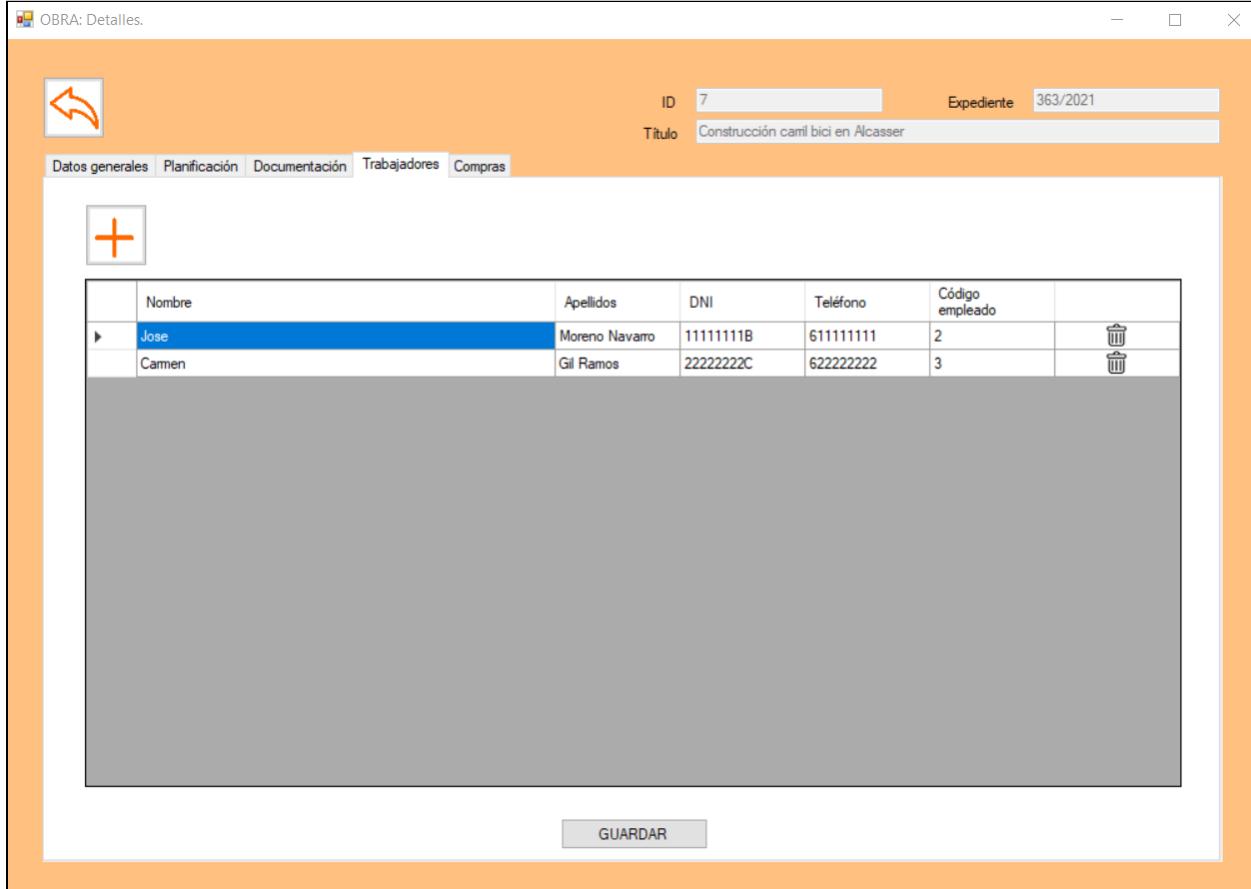
    String SQL = "SELECT * from doc_fin where ID = '" + Variables_Globales.id + "' ORDER BY ID ";
    DataSet ds = Conexiones.Retorna_Datos(SQL);

    string ID = "";
    string LF = "";
    string CF = "";
    string AREC = "";
    string CEE = "";

    ID = Convert.ToString(textBox15.Text);
    LF = checkBox26.Checked.ToString();
    CF = checkBox27.Checked.ToString();
    AREC = checkBox28.Checked.ToString();
    CEE = checkBox29.Checked.ToString();

    if (ds.Tables[0].Rows.Count > 0)
    {
        Conexiones.Ejecuta_Consulta("UPDATE doc_fin set LF = '" + LF + "', CF = '" + CF + "', AREC = '" + AREC +
        "', CEE = '" + CEE + "' where ID = '" + Variables_Globales.id + "'");
    }
    else
    {
        Conexiones.Ejecuta_Consulta("INSERT INTO doc_fin (ID, LF, CF, AREC, CEE) " +
        "VALUES ('" + ID + "', '" + LF + "', '" + CF + "', '" + AREC + "','" + CEE + "')");
    }
}
```

La pestaña de trabajadores permite al usuario asignar o eliminar de obra trabajadores existentes en la BBDD. Por lo tanto, desde esta pestaña no se permite crear nuevos obreros, sólo realizar cambios de los ya existentes. La siguiente imagen muestra en la tabla que la obra tiene asignados a dos trabajadores.



	Nombre	Apellidos	DNI	Teléfono	Código empleado	
▶	Jose	Moreno Navarro	11111111B	611111111	2	
	Carmen	Gil Ramos	22222222C	622222222	3	

En el caso que el usuario decida eliminar a los trabajadores, deberá seleccionar el botón de la papelera cuyo funcionamiento se muestra seguidamente. Se trata de un botón de eliminar “falso”, puesto que no se eliminan todos los datos del empleado de la BBDD, sino que se modifica el estado del trabajador. Se ha creado un “UPDATE” que cambia el valor de la columna “Libre” e introduce un “Sí”, además, cambia la “Obra_asignada” a “00” que es una obra “comodín” que se utiliza para indicar que el trabajador puede asignarse a cualquier otra obra existente.

```

int NumFila = 0;
private void dataGridView3_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    var senderGrid = (DataGridView)sender;

    try
    {
        if (senderGrid.Columns[e.ColumnIndex] is DataGridViewImageColumn && e.RowIndex >= 0)
        {
            NumFila = e.RowIndex;

            //dataGridView3.CurrentCell = dataGridView1[0, NumFila];
        }
    }
}

```

```

        string ID_Eliminar = Convert.ToString(dataGridView3.Rows[NumFila].Cells[4].Value);

        //Eliminar
        if (e.ColumnIndex == 5)
        {
            DialogResult resp = MessageBox.Show("¿Seguro que deseas eliminar al trabajador de esta
obra?", "¡ALERTA!", MessageBoxButtons.YesNo);

            if (resp == DialogResult.Yes)
            {
                string SQL = "UPDATE trabajadores_obra SET Libre = 'SI', Obra_asignada = '00' where
CODI = '" + ID_Eliminar + "'";
                Conexiones.Ejecuta_Consulta("UPDATE trabajadores_obra SET Libre = 'SI',
Obra_asignada = '00' where CODI = '" + ID_Eliminar + "'");

                MessageBox.Show("El trabajador ha quedado LIBRE.");
                Cargar_Grid_Trabajadores();
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error eliminando el registro indicado.");
    }
}

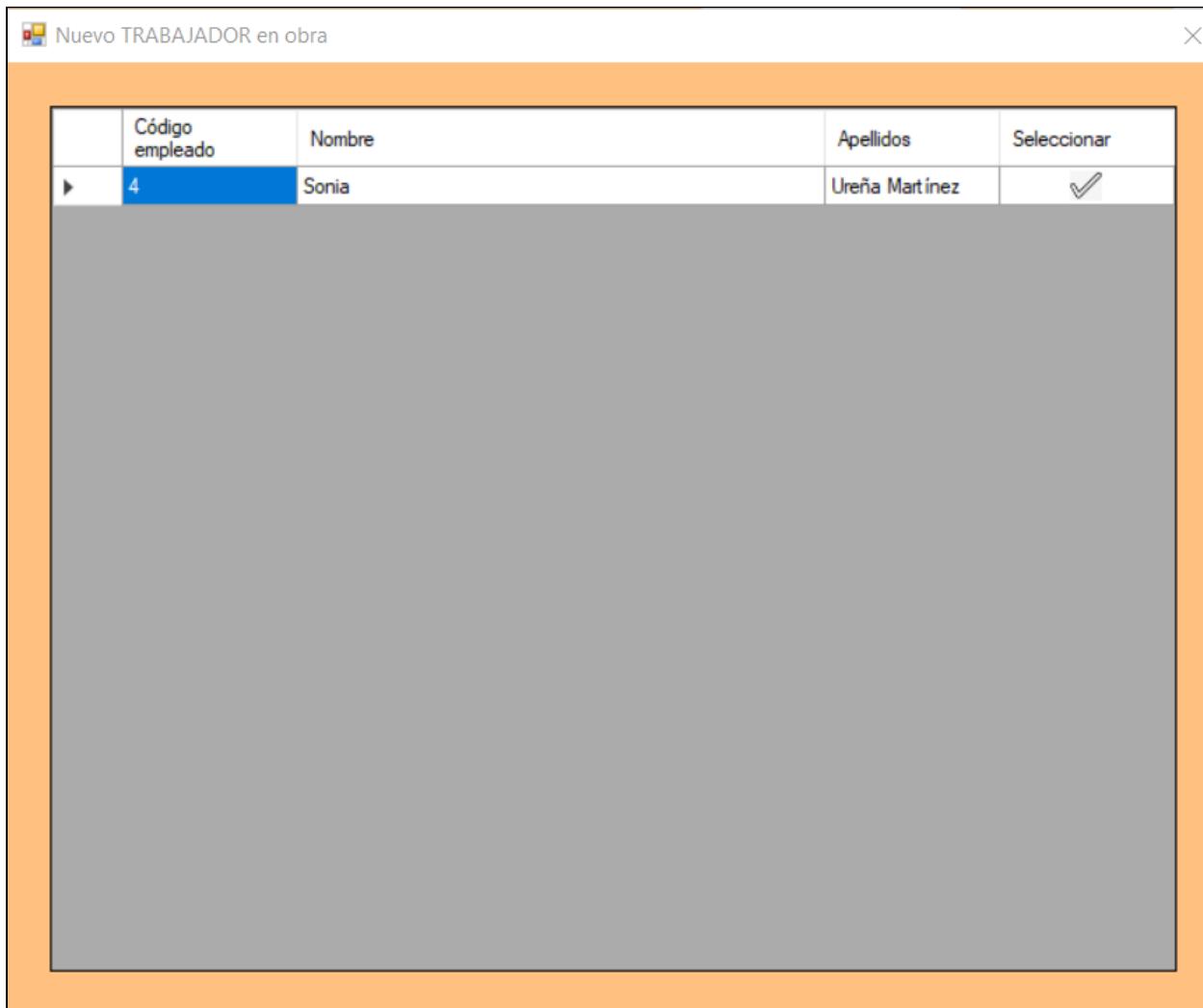
```

El caso contrario se realiza desde el botón “+”, al pulsarlo aparece un listado de los empleados libres. Para mostrar este listado, se ha creado un método que selecciona sólo los trabajadores libres en la BBDD, esto se logra incluyendo un “where” en la consulta de SQL.

```

private void Cargar_Grid_Trabajadores()
{
    string SQL = "SELECT * FROM trabajadores_obra where Libre = 'SI'";
    DataSet ds = Conexiones.Retorna_Datos("SELECT * FROM trabajadores_obra where Libre = 'SI'");
    dataGridView1.AutoGenerateColumns = false;
    dataGridView1.DataSource = ds.Tables[0];
}

```



Para agregar un trabajador libre a la obra actual, se debe pulsar sobre el último botón con una imagen de “tick”. La función es similar a la de eliminar, pero en este caso, se cambia el valor de “Libre” a “NO” y además, se recoge que para ese trabajador la obra asignada será la que se ha definido anteriormente para “Variables_Globales.id”.

```

int NumFila = 0;
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    var senderGrid = (DataGridView)sender;

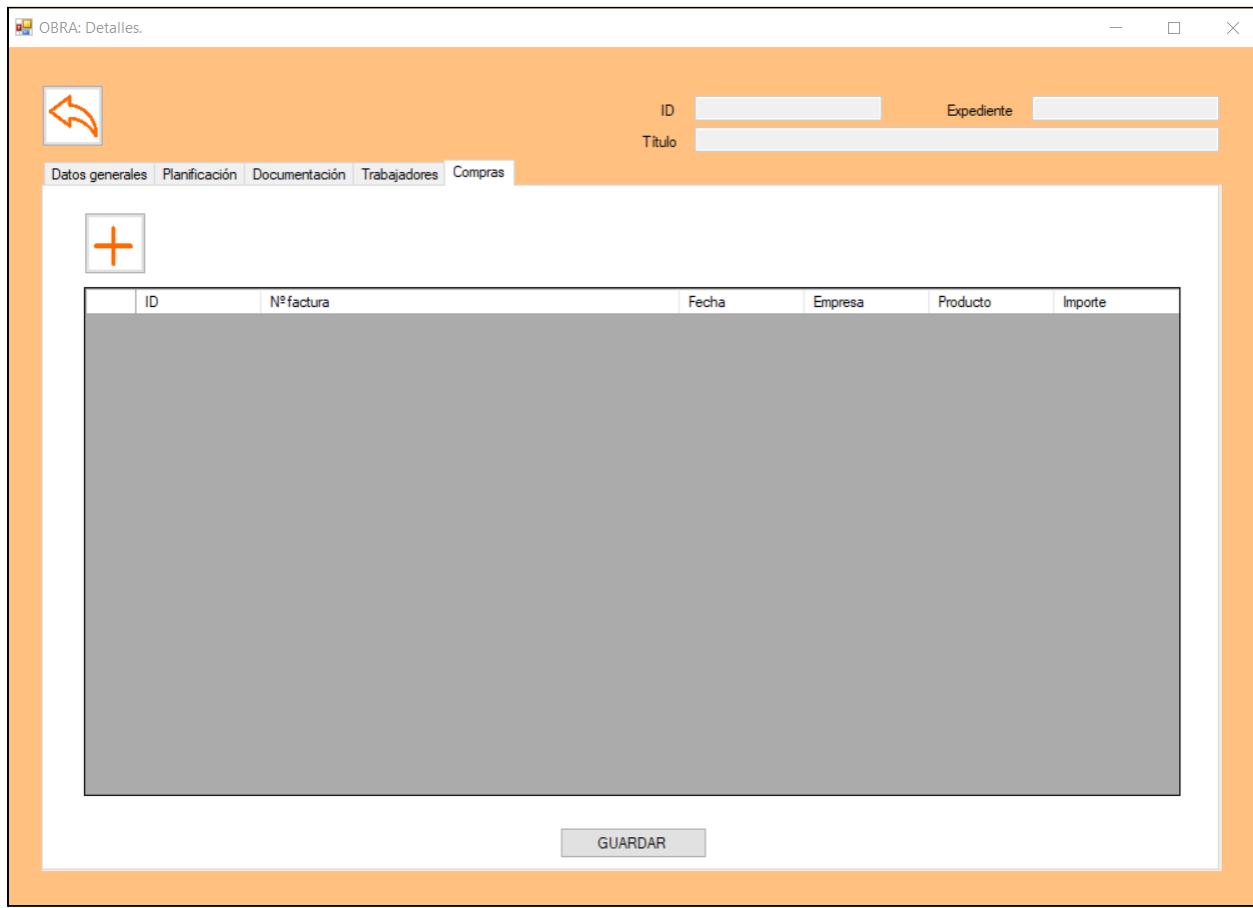
    try
    {
        if (senderGrid.Columns[e.ColumnIndex] is DataGridViewImageColumn && e.RowIndex >= 0)
        {
            NumFila = e.RowIndex;

            dataGridView1.CurrentCell = dataGridView1[0, NumFila];
            string Trabajador = Convert.ToString(dataGridView1.Rows[NumFila].Cells[0].Value);
            //Seleccionar
            if (e.ColumnIndex == 3)
            {
                DialogResult resp = MessageBox.Show("¿Quieres añadir al trabajador a la obra " +
Variables_Globales.id+ "?", "¡ALERTA!", MessageBoxButtons.YesNo);

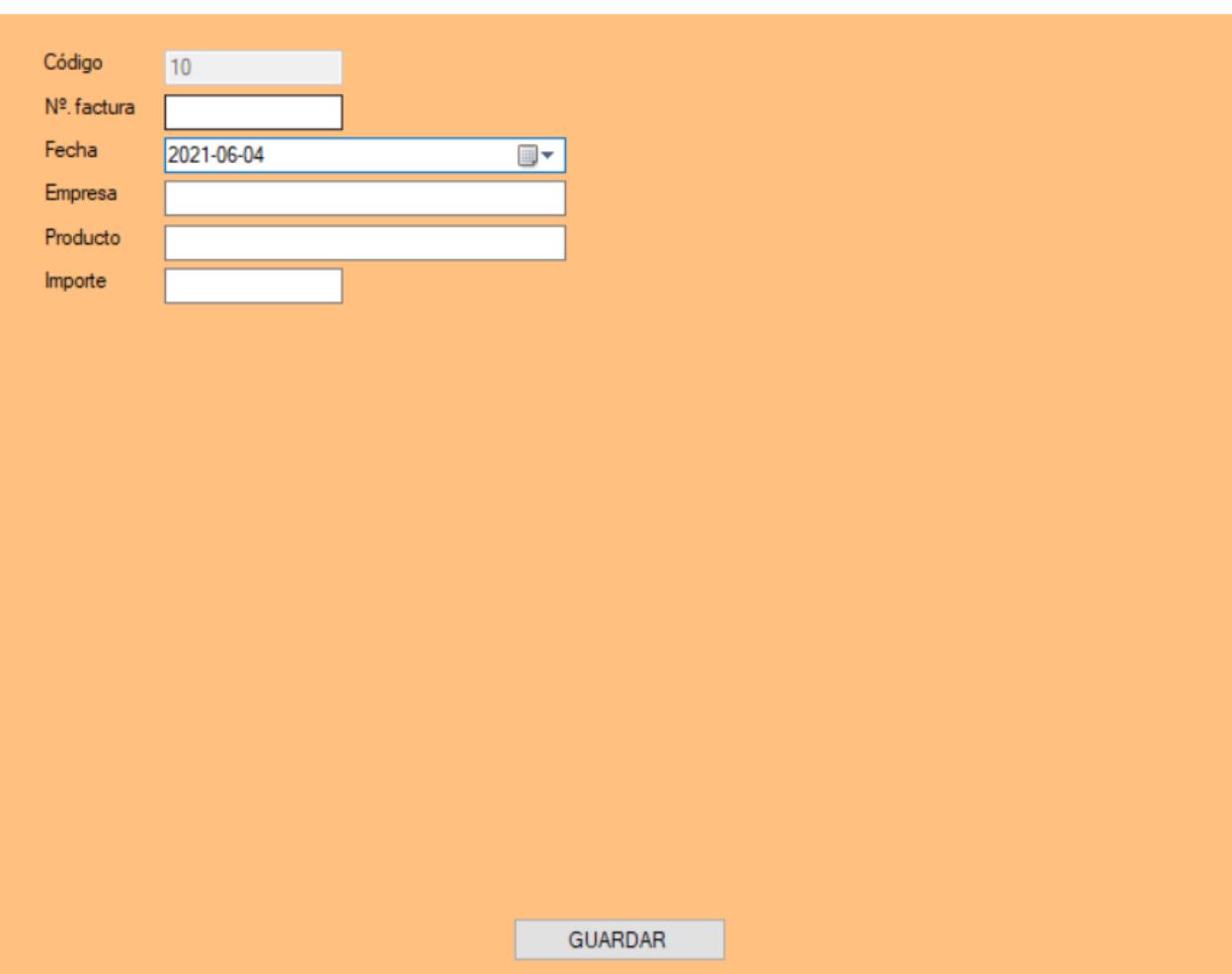
                if (resp == DialogResult.Yes)
                {
                    string SQL = "UPDATE trabajadores_obra SET Libre = 'NO', Obra_asignada = '" +
Variables_Globales.id+ "' WHERE CODI = " + Trabajador + "'";
                    Conexiones.Ejecuta_Consulta("UPDATE trabajadores_obra SET Libre = 'NO', Obra_asignada = '" +
Variables_Globales.id + "' WHERE CODI = " + Trabajador + "'");
                    MessageBox.Show("Añadido correctamente.");
                    this.Close();
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error modificando el registro indicado.");
        }
    }
}

```

La última pestaña es para las compras, el diseño es similar a las anteriores y, al pulsar sobre el ícono “+” permite añadir nuevas compras.



Para anotar nuevas compras se deberán indicar los datos de factura, fecha, empresa, producto e importe. Como se ha visto anteriormente el campo de identificador se autocompleta.



Nueva COMPRA.

Código	10
Nº. factura	
Fecha	2021-06-04
Empresa	
Producto	
Importe	

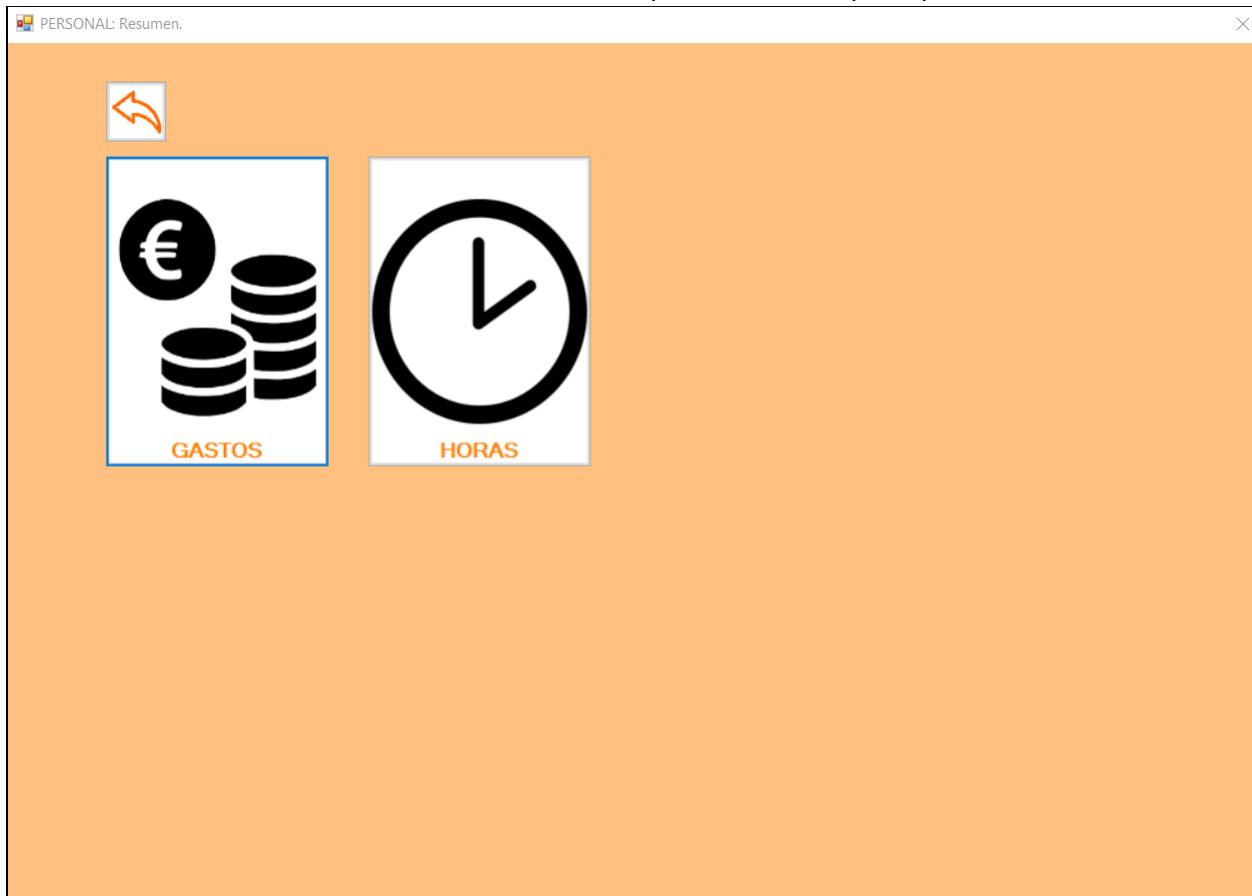
GUARDAR

En el caso de la fecha no se permite que el usuario pueda escribir sobre la caja, por lo que se ha utilizado un “dateTimePicker”. Este campo se ha definido el archivo de diseño del formulario 13 y se han fijado una serie de características. En este caso se ha customizado el formato de fecha, ya que no se ha considerado necesario guardar el día de la semana ni la hora, por lo tanto, sólo se insertará día, mes y año.

```
this.dateTimePicker1 = new System.Windows.Forms.DateTimePicker();

//
// dateTimePicker1
//
this.dateTimePicker1.CustomFormat = "yyyy-MM-dd";
this.dateTimePicker1.Format = System.Windows.Forms.DateTimePickerFormat.Custom;
this.dateTimePicker1.Location = new System.Drawing.Point(127, 85);
this.dateTimePicker1.Margin = new System.Windows.Forms.Padding(4);
this.dateTimePicker1.Name = "dateTimePicker1";
this.dateTimePicker1.Size = new System.Drawing.Size(299, 22);
this.dateTimePicker1.TabIndex = 9;
```

Desde el menú principal, haciendo click en el botón “PERSONAL” se accede a un submenú cuyo diseño contiene dos botones que conducen al usuario a otras ventanas. Las funciones de los botones se han definido de la misma manera que en el menú principal.



Antes de continuar con las acciones que se realizan desde este formulario, se dispone a continuación el archivo de diseño. Como se puede observar se indica que forma parte del “Form5”. Seguidamente se encuentra una parte de código que se genera automáticamente y, luego se crean las variables, en este caso, el formulario cuenta con un panel donde se insertan tres botones, el de salir, el de “GASTOS” y el de “HORAS”.

Más adelante, se define cada uno de estos elementos. Por ejemplo, para el “panel1” se indica la localización, los márgenes, el tamaño, el color, etc. Para cada botón, se indica la imagen que contiene, la localización, el tamaño y la acción “click”, entre otros.

```
namespace Dacobrick
{
    partial class Form5
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// 
```

```

/// </summary>
/// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Form5));
    this.panel1 = new System.Windows.Forms.Panel();
    this.button3 = new System.Windows.Forms.Button();
    this.button2 = new System.Windows.Forms.Button();
    this.button1 = new System.Windows.Forms.Button();
    this.panel1.SuspendLayout();
    this.SuspendLayout();
    //
    // panel1
    //
    this.panel1.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(192)))), ((int)(((byte)(128)))));
    this.panel1.Controls.Add(this.button3);
    this.panel1.Controls.Add(this.button2);
    this.panel1.Controls.Add(this.button1);
    this.panel1.Location = new System.Drawing.Point(0, 0);
    this.panel1.Margin = new System.Windows.Forms.Padding(4);
    this.panel1.Name = "panel1";
    this.panel1.Size = new System.Drawing.Size(1340, 945);
    this.panel1.TabIndex = 0;
    //
    // button3
    //
    this.button3.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("button3.BackgroundImage")));
    this.button3.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch;
    this.button3.Location = new System.Drawing.Point(107, 37);
    this.button3.Margin = new System.Windows.Forms.Padding(4);
    this.button3.Name = "button3";
    this.button3.Size = new System.Drawing.Size(67, 62);
    this.button3.TabIndex = 7;
    this.button3.UseVisualStyleBackColor = true;
    this.button3.Click += new System.EventHandler(this.button3_Click);
    //
    // button2
    //
    this.button2.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)));
    this.button2.BackColor = System.Drawing.Color.White;

```

```

        this.button2.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("button2.BackgroundImage")));
        this.button2.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Zoom;
        this.button2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold);
        this.button2.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)((byte)(128))),
((int)((byte)(0)))));
        this.button2.Location = new System.Drawing.Point(387, 111);
        this.button2.Margin = new System.Windows.Forms.Padding(4);
        this.button2.Name = "button2";
        this.button2.Size = new System.Drawing.Size(240, 308);
        this.button2.TabIndex = 6;
        this.button2.Text = "HORAS";
        this.button2.TextAlign = System.Drawing.ContentAlignment.BottomCenter;
        this.button2.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageAboveText;
        this.button2.UseVisualStyleBackColor = false;
        this.button2.Click += new System.EventHandler(this.button2_Click);
        //
        // button1
        //
        this.button1.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)));
        this.button1.BackColor = System.Drawing.Color.White;
        this.button1.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("button1.BackgroundImage")));
        this.button1.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Zoom;
        this.button1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.button1.ForeColor = System.Drawing.Color.FromArgb(((int)((byte)(255))), ((int)((byte)(128))),
((int)((byte)(0)))));
        this.button1.Location = new System.Drawing.Point(107, 111);
        this.button1.Margin = new System.Windows.Forms.Padding(4);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(240, 308);
        this.button1.TabIndex = 5;
        this.button1.Text = "GASTOS";
        this.button1.TextAlign = System.Drawing.ContentAlignment.BottomCenter;
        this.button1.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageAboveText;
        this.button1.UseVisualStyleBackColor = false;
        this.button1.Click += new System.EventHandler(this.button1_Click);
        //
        // Form5
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.BackColor = System.Drawing.Color.White;
        this.ClientSize = new System.Drawing.Size(1344, 897);
        this.Controls.Add(this.panel1);
        this.Margin = new System.Windows.Forms.Padding(4);
        this.MaximizeBox = false;
        this.MinimizeBox = false;
        this.Name = "Form5";
        this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
        this.Text = "PERSONAL: Resumen.";
        this.panel1.ResumeLayout(false);
        this.ResumeLayout(false);
    }

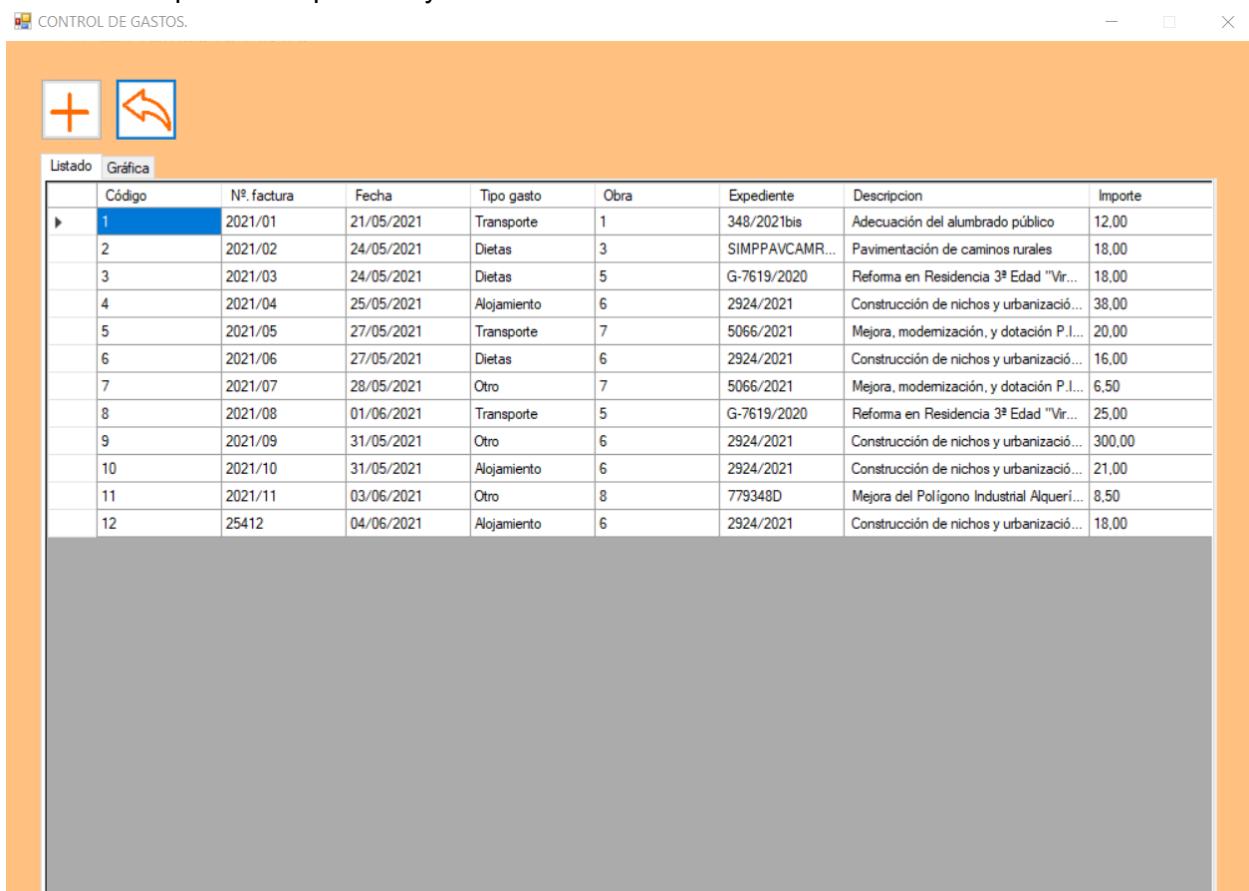
#endregion
}

```

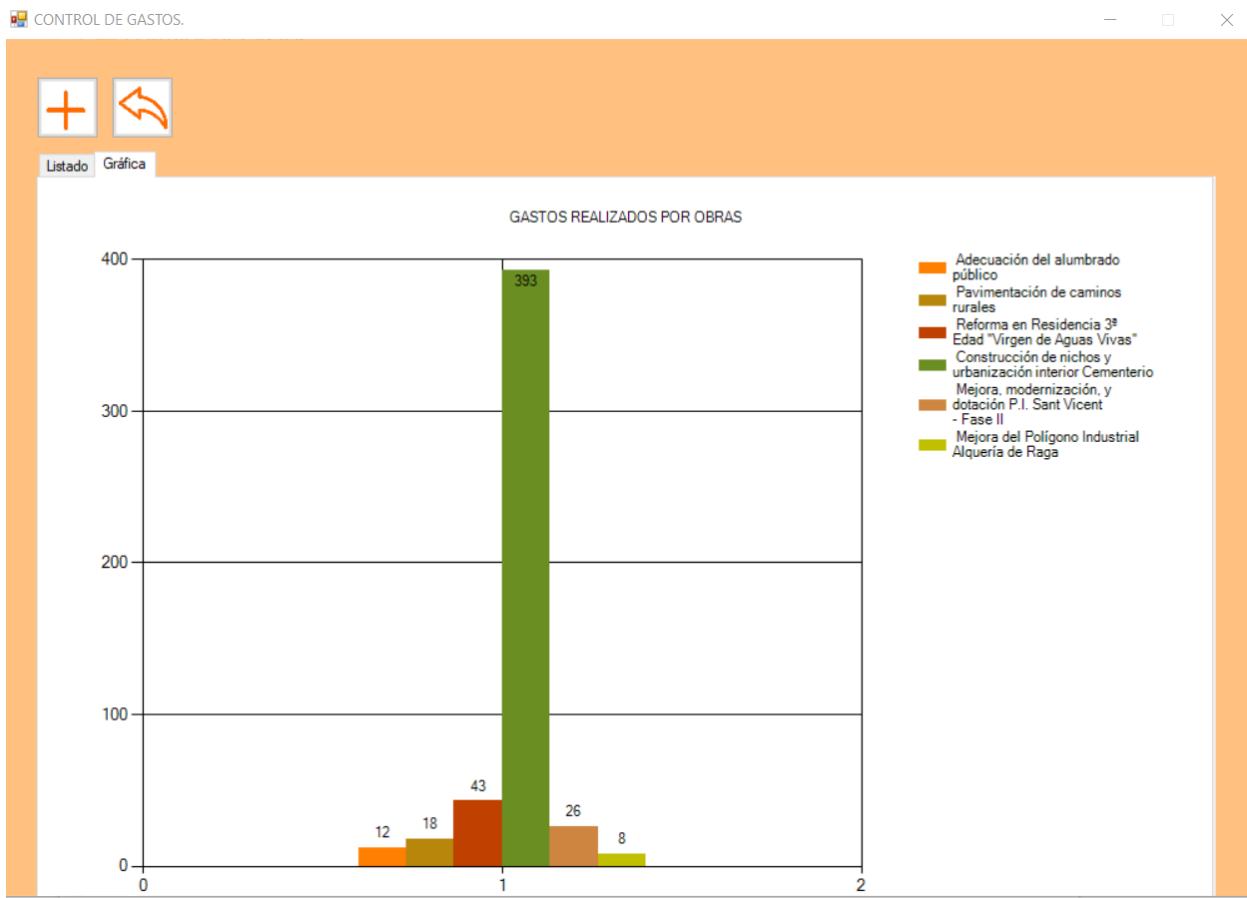
```
private System.Windows.Forms.Panel panel1;
private System.Windows.Forms.Button button1;
private System.Windows.Forms.Button button2;
private System.Windows.Forms.Button button3;
}
}
```

Desde “GASTOS” se abre un formulario que permite al usuario llevar un control de los gastos realizados. Se muestran los iconos de “+” y “atrás” y una ventana con dos pestañas, la primera muestra los gastos en un listado y, la segunda en una gráfica.

Para cargar los datos de la gráfica se ha utilizado una consulta en la BBDD y los resultados devueltos se presentan por filas y columnas en la tabla.



Para el gráfico se ha realizado una consulta a la BBDD que devuelve las cantidades indicadas de los gastos sumados por obra, es decir, el total. Esta pantalla muestra el título de la gráfica en la parte superior, la leyenda en la parte derecha que se compone de las distintas descripciones de obras y, en la parte central la gráfica con los valores totales.



La función que permite visualizar los datos en modo gráfico es “Cargar_Chart()”, para ello se realiza una consulta SQL sumando el importe de los gastos según el título de la obra. Posteriormente, se crean los elementos del “chart” o gráfico, el título, los datos de la leyenda y el total de cada barra.

```

private void Cargar_Chart()
{
    //String SQL = "SELECT YEAR(Fecha) AS yyyy, MONTHNAME(Fecha) as mm, SUM(Importe) AS
    Su FROM gastos " +
    // "GROUP BY YEAR(Fecha), MONTH(Fecha)";

    String SQL = "SELECT TITULO, SUM(Importe) AS Su FROM gastos GROUP BY TITULO";
    DataSet ds = Conexiones.Retorna_Datos(SQL);

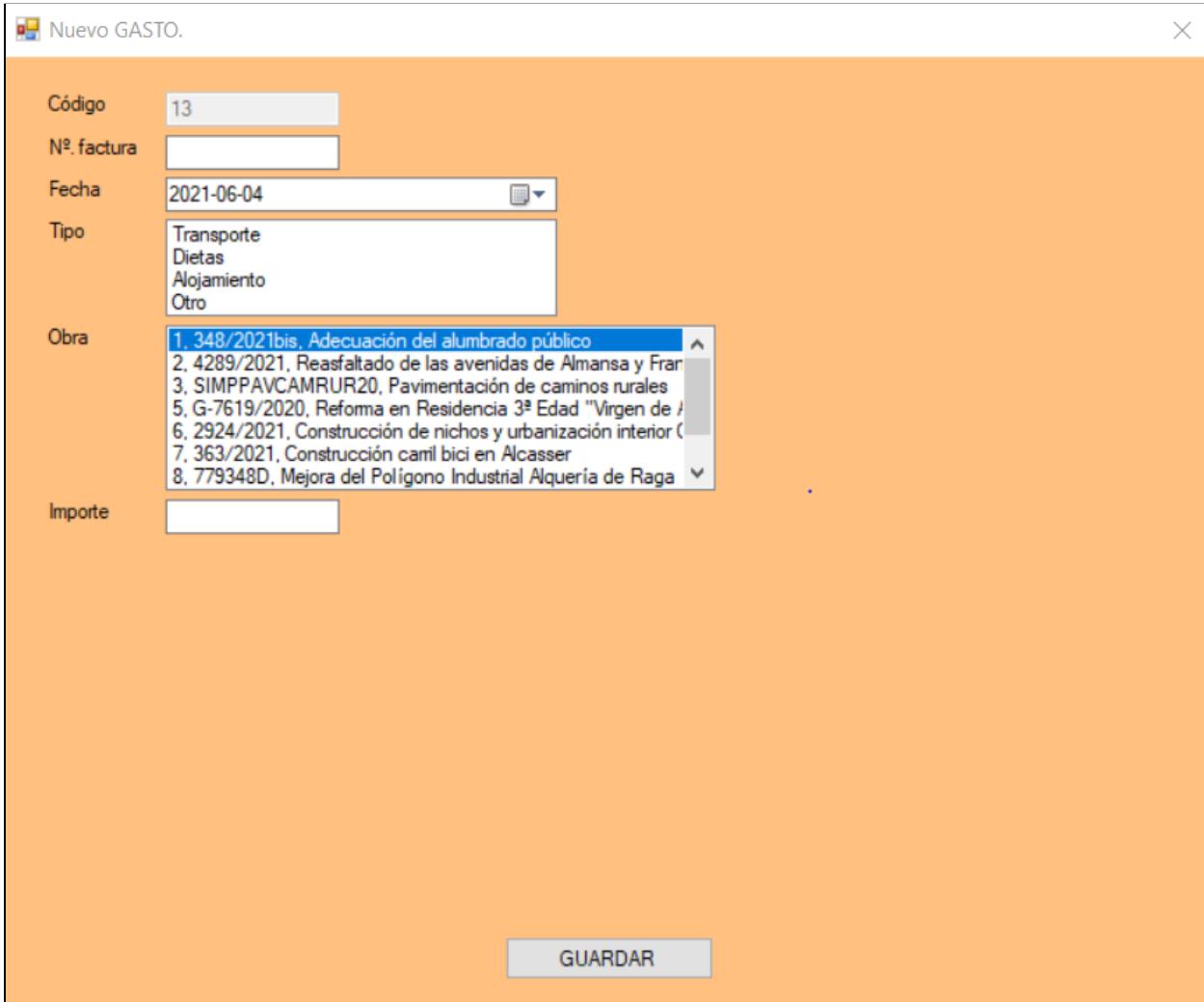
    DataTable firstTable = ds.Tables[0];
    chart1.Series.Clear();

    chart1.Titles.Add("GASTOS REALIZADOS POR OBRAS");
    foreach (DataRow row in firstTable.Rows)
    {
        Series series = chart1.Series.Add(row["TITULO"].ToString());
        series.Points.Add(Convert.ToDouble(row["Su"].ToString()));
        series.Label = row["Su"].ToString();
    }
}

```

}

Para determinar un nuevo gasto se utiliza el siguiente formulario. En el que se recoge el nº de factura, la fecha, el tipo de gasto, la obra a la que pertenece y su importe.



Nuevo GASTO.	
Código	13
Nº. factura	<input type="text"/>
Fecha	2021-06-04 <input type="button" value="..."/>
Tipo	Transporte Dietas Alojamiento Otro
Obra	1, 348/2021bis, Adecuación del alumbrado público 2, 4289/2021, Reasfaltado de las avenidas de Almansa y Frar 3, SIMPPAVCAMRUR20, Pavimentación de caminos rurales 5, G-7619/2020, Reforma en Residencia 3 ^a Edad "Virgen de / 6, 2924/2021, Construcción de nichos y urbanización interior / 7, 363/2021, Construcción caril bici en Alcasser 8, 779348D, Mejora del Polígono Industrial Alqueria de Raga
Importe	<input type="text"/>
<input type="button" value="GUARDAR"/>	

Para el tipo de gasto se ha utilizado un elemento “listBox” en la que se han editado sólo 4 elementos (Transporte, dietas, alojamiento y otro), esto se visualiza en el archivo de diseño de la siguiente forma:

```

this.listBox1.FormattingEnabled = true;
this.listBox1.ItemHeight = 16;
this.listBox1.Items.AddRange(new object[] {
  "Transporte",
  "Dietas",
  "Alojamiento",
  "Otro"});
this.listBox1.Location = new System.Drawing.Point(127, 115);
this.listBox1.Margin = new System.Windows.Forms.Padding(4);
this.listBox1.Name = "listBox1";
this.listBox1.Size = new System.Drawing.Size(299, 68);
this.listBox1.TabIndex = 15;
  
```

Más adelante, en el apartado de problemas encontrados y sus soluciones, se explicará el formato del segundo “listBox” que unifica el identificador, el expediente y el título de cada obra.

Dentro del submenú de “HORAS” se accede a una pantalla muy similar a la anterior, con una pestaña para listado y otra para gráfica. En este caso, se procede a explicar algunos de los elementos del archivo de diseño, este panel contiene un “tabControl” que permite tener varias pestañas en una misma ventana, su definición muestra cómo contiene dos pestañas, su localización, el nombre, tamaño, etc.

```
// tabControl1
//
this.tabControl1.Controls.Add(this.tabPage1);
this.tabControl1.Controls.Add(this.tabPage2);
this.tabControl1.Location = new System.Drawing.Point(40, 111);
this.tabControl1.Margin = new System.Windows.Forms.Padding(4);
this.tabControl1.Name = "tabControl1";
this.tabControl1.SelectedIndex = 0;
this.tabControl1.Size = new System.Drawing.Size(1260, 775);
this.tabControl1.TabIndex = 0;
```

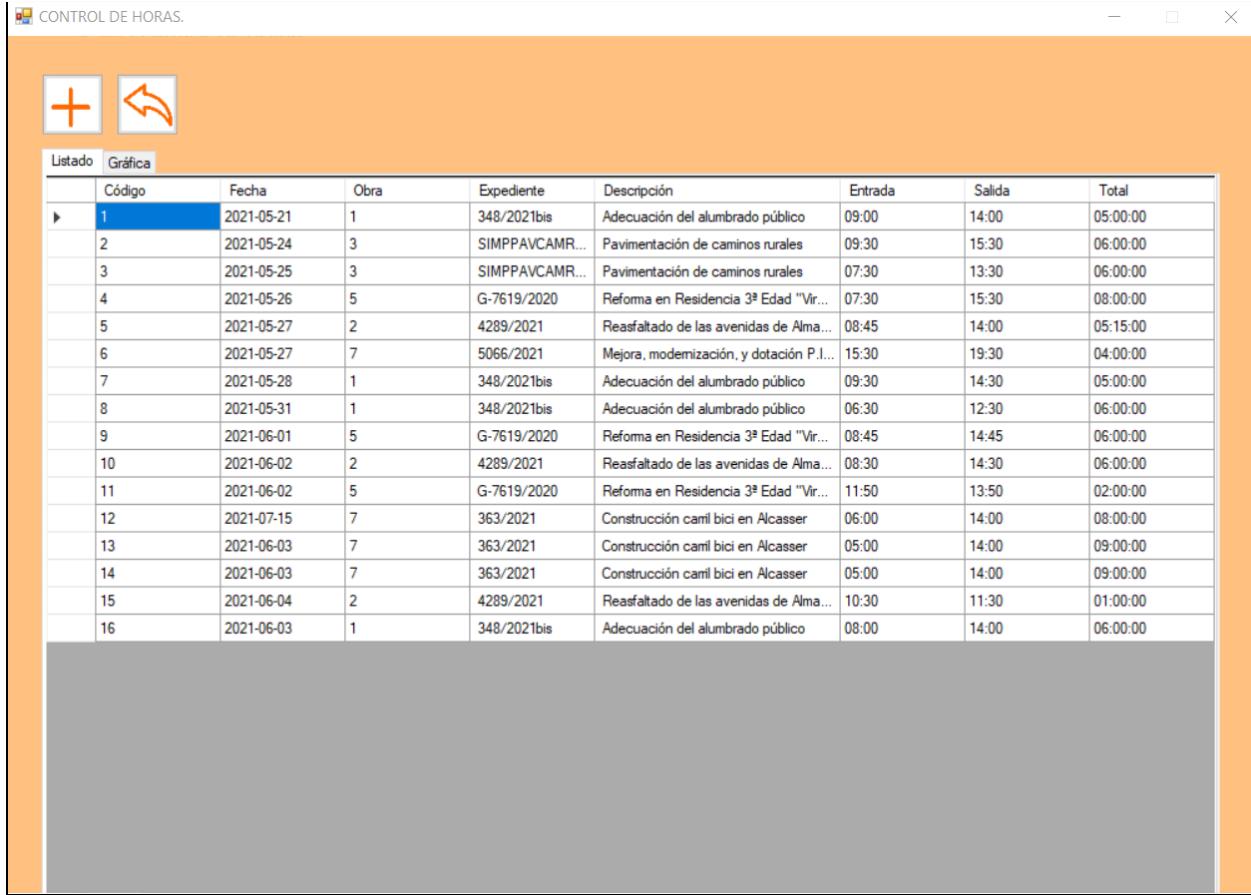
Cada pestaña es un elemento “tabPage” con la siguiente estructura. Por ejemplo, la primera incluye un “dataGridView”, se definen los márgenes, el tamaño, su nombre, etc.

```
// tabPage1
//
this.tabPage1.Controls.Add(this.dataGridView1);
this.tabPage1.Location = new System.Drawing.Point(4, 25);
this.tabPage1.Margin = new System.Windows.Forms.Padding(4);
this.tabPage1.Name = "tabPage1";
this.tabPage1.Padding = new System.Windows.Forms.Padding(4);
this.tabPage1.Size = new System.Drawing.Size(1252, 746);
this.tabPage1.TabIndex = 0;
this.tabPage1.Text = "Listado";
this.tabPage1.UseVisualStyleBackColor = true;
```

Todas las tablas, “dataGridView” de la aplicación tienen una diseño similar, en este caso, se ha indicado que el usuario no tiene permiso ni para añadir ni para eliminar filas. También se ha indicado que las columnas pueden cambiar su tamaño. Posteriormente, se indica el orden y nombre de las columnas y otros aspectos como tamaño, nombre, posición, etc.

```
// dataGridView1
//
this.dataGridView1.AllowUserToAddRows = false;
this.dataGridView1.AllowUserToDeleteRows = false;
this.dataGridView1.ColumnHeadersHeightSizeMode = System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.ColumnHeadersHeightSizeMode.AutoSize;
this.dataGridView1.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
this.Codigo,
this.Fecha,
this.Obra,
this.Expediente,
this.Descripción,
this.Entrada,
this.Salida,
this.Total});
this.dataGridView1.Location = new System.Drawing.Point(-3, 0);
```

```
this.dataGridView1.Margin = new System.Windows.Forms.Padding(4);
this.dataGridView1.Name = "dataGridView1";
this.dataGridView1.Size = new System.Drawing.Size(1253, 745);
this.dataGridView1.TabIndex = 1;
```

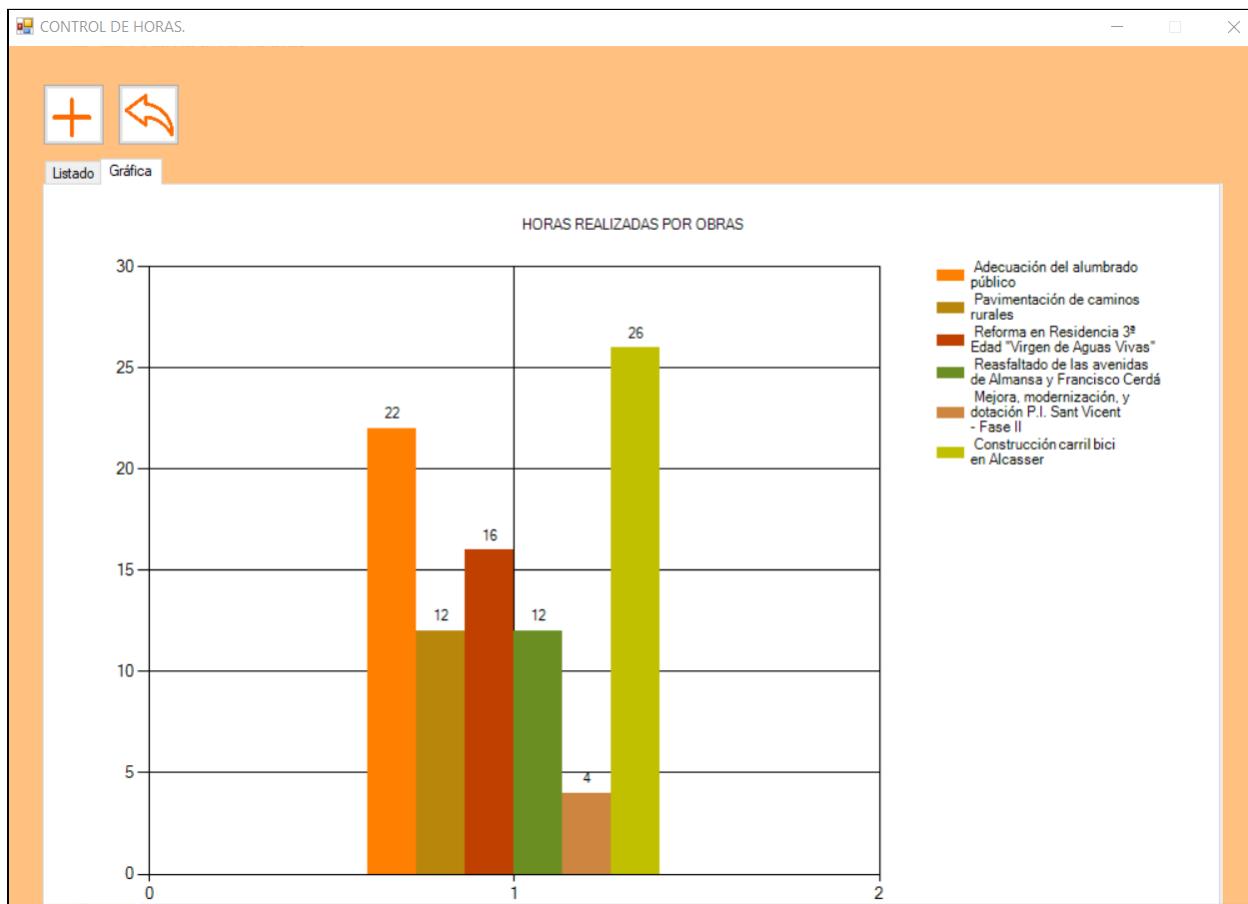


The screenshot shows a Windows application window titled "CONTROL DE HORAS". At the top left are two buttons: a red square with a white plus sign and a blue square with a white left-pointing arrow. Below them is a tab bar with "Listado" selected and "Gráfica" as the other tab. The main area contains a DataGridView with the following data:

Código	Fecha	Obra	Expediente	Descripción	Entrada	Salida	Total
1	2021-05-21	1	348/2021bis	Adecuación del alumbrado público	09:00	14:00	05:00:00
2	2021-05-24	3	SIMPPAVCAMR...	Pavimentación de caminos rurales	09:30	15:30	06:00:00
3	2021-05-25	3	SIMPPAVCAMR...	Pavimentación de caminos rurales	07:30	13:30	06:00:00
4	2021-05-26	5	G-7619/2020	Reforma en Residencia 3 ^a Edad "Vir..."	07:30	15:30	08:00:00
5	2021-05-27	2	4289/2021	ReASFALTADO de las avenidas de Alma...	08:45	14:00	05:15:00
6	2021-05-27	7	5066/2021	Mejora, modernización, y dotación P.I...	15:30	19:30	04:00:00
7	2021-05-28	1	348/2021bis	Adecuación del alumbrado público	09:30	14:30	05:00:00
8	2021-05-31	1	348/2021bis	Adecuación del alumbrado público	06:30	12:30	06:00:00
9	2021-06-01	5	G-7619/2020	Reforma en Residencia 3 ^a Edad "Vir..."	08:45	14:45	06:00:00
10	2021-06-02	2	4289/2021	ReASFALTADO de las avenidas de Alma...	08:30	14:30	06:00:00
11	2021-06-02	5	G-7619/2020	Reforma en Residencia 3 ^a Edad "Vir..."	11:50	13:50	02:00:00
12	2021-07-15	7	363/2021	Construcción camí bici en Alcasser	06:00	14:00	08:00:00
13	2021-06-03	7	363/2021	Construcción camí bici en Alcasser	05:00	14:00	09:00:00
14	2021-06-03	7	363/2021	Construcción camí bici en Alcasser	05:00	14:00	09:00:00
15	2021-06-04	2	4289/2021	ReASFALTADO de las avenidas de Alma...	10:30	11:30	01:00:00
16	2021-06-03	1	348/2021bis	Adecuación del alumbrado público	08:00	14:00	06:00:00

Otro elemento de este formulario es la gráfica, cuyo diseño contiene su localización, nombre, tamaño y la paleta de colores utilizados, entre otros elementos.

```
// chart1
//
chartArea1.Name = "ChartArea1";
this.chart1.ChartAreas.Add(chartArea1);
legend1.Name = "Legend1";
this.chart1.Legends.Add(legend1);
this.chart1.Location = new System.Drawing.Point(7, 7);
this.chart1.Name = "chart1";
this.chart1.Palette = System.Windows.Forms.DataVisualization.Charting.ChartColorPalette.EarthTones;
this.chart1.Size = new System.Drawing.Size(1238, 732);
this.chart1.TabIndex = 0;
this.chart1.Text = "chart1";
```



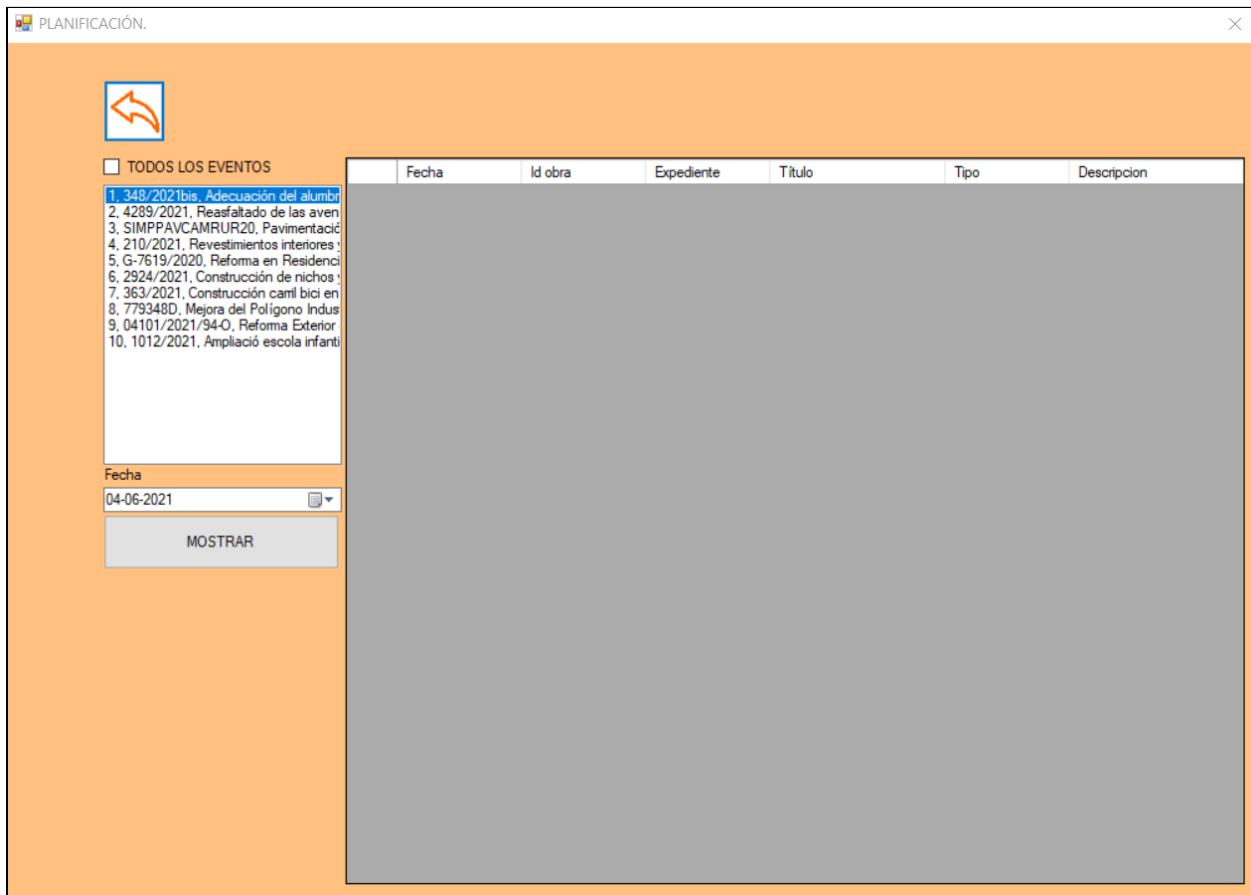
La única diferencia en este punto del proyecto se encuentra a la hora de introducir los datos. Para el correcto registro en la BBDD se debe recoger en este formulario: la fecha, la obra y las horas de entrada y de salida. Se ha creado una fórmula que calcula una vez introducidos los datos, el total de horas dedicadas, esta se explica en el apartado de problemas encontrados y sus soluciones. El botón “GUARDAR” recoge todos los datos y realiza un “INSERT INTO” en la tabla correspondiente.

Nuevo registro HORAS.

Código	17
Fecha	2021-06-04
Obra	1. 348/2021bis. Adecuación del alumbrado público 2. 4289/2021. Reasfaltado de las avenidas de Almansa y Fran 3. SIMPPAVCAMRUR20. Pavimentación de caminos rurales 5. G-7619/2020. Reforma en Residencia 3 ^a Edad "Virgen de / 6. 2924/2021. Construcción de nichos y urbanización interior C 7. 363/2021. Construcción caril bici en Alcasser 8. 779348D. Mejora del Polígono Industrial Alquería de Raga
H. Entrada	<input type="text"/>
H. Salida	<input type="text"/>
Total	<input type="text"/>

GUARDAR

Desde el menú principal, pulsando sobre “PLANIFICACIÓN” el usuario puede acceder a un formulario que le permite revisar los eventos agendados. La función de dicho formulario es indicar si se quiere ver los eventos de todas las obras o exclusivamente los relacionados con una y la fecha muestra en la parte de la derecha dichos eventos.



Esta ventana tiene un “checkbox” que si se pulsa anula la selección en la lista inferior, esto se ha conseguido con el siguiente método. Si se pulsa sobre el “checkbox” la lista pasa a estar desactivada.

```
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    if (checkBox1.Checked == true)
    {
        listBox1.Enabled = false;
    }
    else if (checkBox1.Checked == false)
    {
        listBox1.Enabled = true;
    }
}
```

Por su parte, el botón mostrar realiza la siguiente operación. Primero comprueba si está seleccionado el “checkbox” de “TODOS LOS EVENTOS”, si es así comprueba que también se ha seleccionado una fecha, en ese caso, realiza una consulta a la base de datos para listar todos los eventos de la fecha seleccionada.

En el caso de que el botón de “TODOS LOS EVENTOS” no esté seleccionado, se recogerá la obra seleccionada del listado y la fecha, con estos datos se consulta en la BBDD la lista de eventos para dicha obra y fecha y se mostrará en el “dataGridView”.

Además, se ha configurado el método para que muestre una serie de errores, si el botón “TODOS LOS EVENTOS” no está seleccionado ni tampoco se ha seleccionado una obra del listado, mostrará una aviso al usuario para que se seleccione la/s obra/s. También aparecerá un mensaje similar en el caso de que no se seleccione ninguna fecha.

Por último, si no existen eventos para la obra/s y/o fecha indicada se mostrará un mensaje informando al usuario.

```

private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.DataSource = "";

    if (checkBox1.Checked == true)
    {
        if(dateTimePicker1.Text != "")
        {
            string Fecha_inicio = Convert.ToString(dateTimePicker1.Text);
            //string Fecha_fin = Convert.ToString(dateTimePicker2.Text);
            //String SQL = "SELECT * FROM planificacion inner join obras on planificacion.ID_obra =
            obras.ID" +
            // " where Fecha between " + Fecha_desde + " and " + Fecha_hasta + "";
            String SQL = "SELECT * FROM planificacion inner join obras on planificacion.ID_obra =
            obras.ID" +
            " where Fecha = " + Fecha_desde + "";
            DataSet ds = Conexiones.Retorna_Datos(SQL);

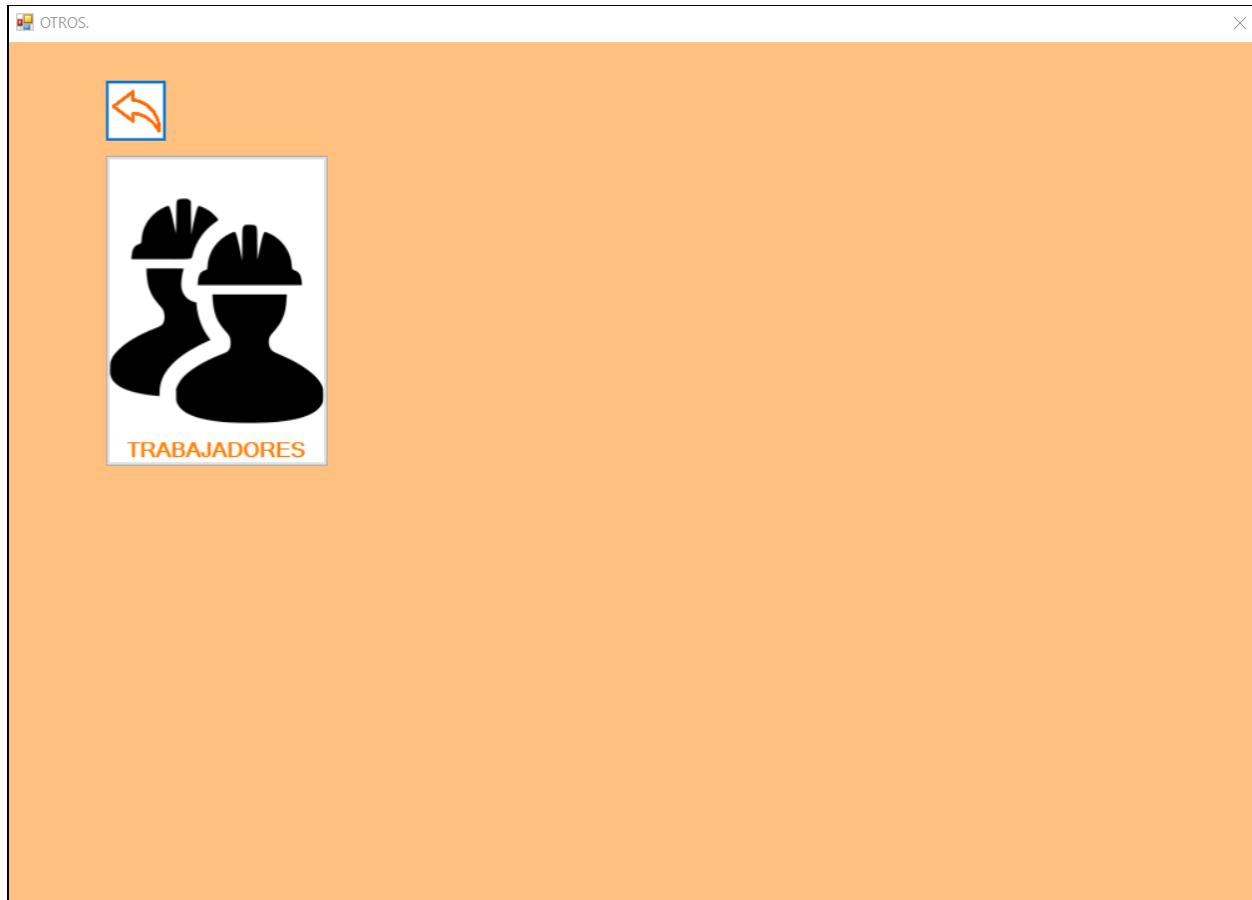
            if (ds.Tables[0].Rows.Count > 0)
            {
                dataGridView1.AutoGenerateColumns = false;
                dataGridView1.DataSource = ds.Tables[0];
            }
            else
            {
                MessageBox.Show("No hay registros para la/s obra/s y/o fecha/s seleccionada/s.");
            }
        }
        else
        {
            MessageBox.Show("Debes seleccionar FECHA.");
        }
    }
    else if (listBox1.Text != "")
    {
        if (dateTimePicker1.Text != "")
        {
            string Fecha_inicio = Convert.ToString(dateTimePicker1.Text);
            //string Fecha_fin = Convert.ToString(dateTimePicker2.Text);
            //String SQL = "SELECT * FROM planificacion inner join obras on planificacion.ID_obra =
            obras.ID" +
            // " where planificacion.ID_obra = " + ID + " and Fecha between " + Fecha_desde + " and " +
            // Fecha_hasta + "";
            String SQL = "SELECT * FROM planificacion inner join obras on planificacion.ID_obra =
            obras.ID" +
            " where planificacion.ID_obra = " + ID + " and Fecha = " + Fecha_desde + "";
        }
    }
}

```

```
DataSet ds = Conexiones.Retorna_Datos(SQL);

if (ds.Tables[0].Rows.Count > 0)
{
    dataGridView1.AutoGenerateColumns = false;
    dataGridView1.DataSource = ds.Tables[0];
}
else
{
    MessageBox.Show("No hay registros para la/s obra/s y/o fecha/s seleccionada/s.");
}
else
{
    MessageBox.Show("Debes seleccionar FECHA.");
}
else
{
    MessageBox.Show("Debes seleccionar la/s obra/s.");
}
```

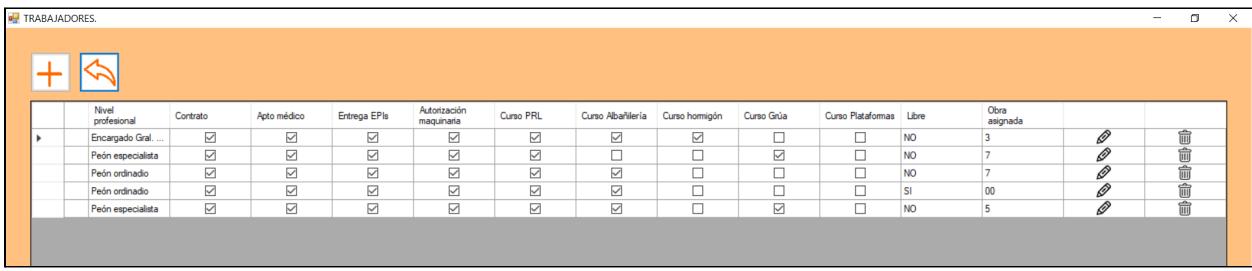
El último botón del menú principal da acceso a un submenú de “TRABAJADORES”:



Cuando se hace accede a “TRABAJADORES” aparecen los botones para añadir y para volver atrás. También un listado de resumen de los empleados dados de alta, en este caso se ha considerado útil presentar toda la información de cada uno de ellos, desde la información personal hasta los cursos que han realizado. Además, para cada trabajador y, al final de cada fila del listado, existe la posibilidad de modificarlo o eliminarlo.



Código empleado	Nombre	Apellidos	DNI	Teléfono	Nivel profesional	Contrato	Apto médico	Entrega EPIs	Autorización maquinaria	Curso PRL	Curso Albañilería	Curso hormigón	Curso Grúa	Curse
1	Manuel	García López	00000000A	600000000	Encargado Gral...	<input checked="" type="checkbox"/>								
2	Jose	Moreno Navarro	11111111B	611111111	Peón especialista	<input checked="" type="checkbox"/>								
3	Carmen	Gil Ramos	22222222C	622222222	Peón ordinario	<input checked="" type="checkbox"/>								
4	Sonia	Ureña Martínez	33333333D	633333333	Peón ordinario	<input checked="" type="checkbox"/>								
5	Pepe	dasd	0000000	6666	Peón especialista	<input checked="" type="checkbox"/>								



Nivel profesional	Contrato	Apto médico	Entrega EPIs	Autorización maquinaria	Curso PRL	Curso Albañilería	Curso hormigón	Curso Grúa	Curso Plataformas Libre	Obra asignada			
Encargado Gral ...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NO	3									
Peón especialista	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NO	7									
Peón ordinario	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NO	7									
Peón ordinario	<input checked="" type="checkbox"/>	<input type="checkbox"/>	SI	00									
Peón especialista	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NO	5									

Los datos se muestran en el listado tras realizar una consulta en la BBDD, al igual que en los “dataGridView” anteriores.

```
private void Cargar_Grid_Trabajadores()
{
    DataSet ds = Conexiones.Retorna_Datos("SELECT * FROM trabajadores_obra ORDER BY CODI");
    dataGridView1.AutoGenerateColumns = false;
    dataGridView1.DataSource = ds.Tables[0];
}
```

He creado un método que captura cuándo el usuario realiza un click en el listado, si se pulsa sobre la columna 18 se ejecuta el método para eliminar al trabajador de la BBDD. Primero, se pregunta al usuario si quiere eliminar definitivamente al trabajador, si la respuesta es afirmativa se envía una sentencia SQL indicando la tabla y el empleado a eliminar. En cambio, si se pulsa sobre la columna 17, se ejecuta el método de editar, por lo tanto, se abre una nueva pantalla.

```
int NumFila = 0;
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    var senderGrid = (DataGridView)sender;

    try
    {
        if (senderGrid.Columns[e.ColumnIndex] is DataGridViewImageColumn && e.RowIndex >= 0)
        {
            NumFila = e.RowIndex;

            dataGridView1.CurrentCell = dataGridView1[0, NumFila];
            string ID_Eliminar = Convert.ToString(dataGridView1.Rows[NumFila].Cells[0].Value);
            //Eliminar
            if (e.ColumnIndex == 18)
```

```

    }

    DialogResult resp = MessageBox.Show("¿Seguro que deseas eliminar el registro?", "¡ALERTA!",
MessageBoxButtons.YesNo);

    if (resp == DialogResult.Yes)
    {
        string SQL = "DELETE FROM trabajadores_obra where CODI = '"+ID_Eliminar+"'";
        Conexiones.Ejecuta_Consulta("DELETE FROM trabajadores_obra where CODI = '"+ID_Eliminar+"');

        MessageBox.Show("Eliminado correctamente.");
        Cargar_Grid_Trabajadores();
        //this.Close();
    }
}

//Editar
if (e.ColumnIndex == 17)
{
    string codigo_empleado = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();

    Variables_Globales.codigo_empleado = codigo_empleado;
    Variables_Globales.Form16_Desde = "FORM10";

    Form16 frm = new Form16();
    frm.ShowDialog();
    Cargar_Grid_Trabajadores();

    //Variables_Globales.id = "";
    //Variables_Globales.expediente = "";
    //Variables_Globales.titulo = "";
    Variables_Globales.Form16_Desde = "";

}

}

}

catch (Exception ex)
{
    MessageBox.Show("Error eliminando el registro indicado.");
}
}

```

La pantalla que se abre es la misma que se utiliza para crear nuevos trabajadores y tiene el siguiente diseño. En ella se recogen datos personales, se indica si el trabajador está o no asignado a una obra, la documentación y los cursos de los que dispone. En el caso de "Editar" algunos de los campos aparecerán bloqueados.

Nuevo TRABAJADOR

Código	5	Apellidos	dasd
Nombre	Pepe	LIBRE	<input checked="" type="checkbox"/> SI <input type="checkbox"/> NO
DNI	0000000	Obra	1, 348/2021bis, Adecuación del alumbrado 2, 4289/2021, Reasfaltado de las avenidas 3, SIMPPAVCAMRUR20, Pavimentación d 4, 210/2021, Revestimientos interiores y ur 5, G-7619/2020, Reforma en Residencia 3 6, 2924/2021, Construcción de nichos y ur 7, 363/2021, Construcción camí bici en Alc
Teléfono	6666		
Nivel prof.	Encargado Gral. Obra Oficial 1 ^ª Oficial 2 ^ª Peón especialista Peón ordinario Aprendiz		
Documentación	<input checked="" type="checkbox"/> Contrato de trabajo firmado <input checked="" type="checkbox"/> Entrega de EPI's <input checked="" type="checkbox"/> Autorización uso maquinaria <input type="checkbox"/> APTO revisión médica		
Cursos realizados	<input checked="" type="checkbox"/> PRL <input checked="" type="checkbox"/> Grúa <input checked="" type="checkbox"/> Albañilería <input type="checkbox"/> Plataformas <input type="checkbox"/> Hormigón		
5	G-7619/2020	Reforma en Reside	GUARDAR

Una vez se inserten o se modifiquen los datos, el usuario deberá pulsar sobre “GUARDAR”, en este momento, se ejecuta la siguiente función. Primero se han creado las distintas variables y se indica desde dónde se deben completar los datos. Seguidamente, se asigna por defecto el valor “00” a la obra si el trabajador está libre, en caso contrario, se recogen los datos de la obra indicada.

Si todos los datos que no deben ser nulos tienen un valor asignado, se realiza una consulta a la BBDD para comprobar si ese trabajador ya existe o no. En el caso de que exista, se realiza una actualización, para ello se envía una consulta SQL de “UPDATE” con la tabla a actualizar, el identificador del trabajador y los datos a actualizar. En caso contrario, se trata de un trabajador nuevo por lo que se realiza un “INSERT”. Finalmente, se ha creado un mensaje para informar al usuario que debe introducir los datos básicos, en caso contrario no se realizarán ni modificaciones ni creaciones.

```
private void button4_Click(object sender, EventArgs e)
{
  string CODI = "";
  string Nombre = "";
```

```

string Apellidos = "";
string DNI = "";
string Telefono = "";
string Nivel = "";
string Contrato = "";
string EPIs = "";
string Maquinaria = "";
string Apto = "";
string PRL = "";
string Albañileria = "";
string Hormigon = "";
string Grua = "";
string Plataformas = "";
string Libre = "";
string Obra = "";

CODI = textBox1.Text;
Nombre = textBox2.Text;
Apellidos = textBox5.Text;
DNI = textBox3.Text;
Telefono = textBox4.Text;
Nivel = listBox1.Text;
Contrato = checkBox1.Checked.ToString();
EPIs = checkBox2.Checked.ToString();
Maquinaria = checkBox3.Checked.ToString();
Apto = checkBox9.Checked.ToString();
PRL = checkBox4.Checked.ToString();
Albañileria = checkBox5.Checked.ToString();
Hormigon = checkBox6.Checked.ToString();
Grua = checkBox7.Checked.ToString();
Plataformas = checkBox8.Checked.ToString();
Libre = listBox2.Text;

if (listBox2.Text == "SI")
{
    Obra = "00";
}
else if (listBox2.Text == "NO")
{
    Obra = textBox_ID.Text;
}

if (Libre == "NO")
{
    if (listBox3.SelectedIndex == 0)
    {
        MessageBox.Show("Si el trabajador NO está libre, debes asignarlo a una obra.");
    }
    else
    {
        if (Convert.ToString(CODI) != "" && Nombre != "" && DNI != "" && Telefono != "" && Libre != "")
        {
            string SQL = "Select * from trabajadores_obra where CODI = " + CODI + "";
            DataSet ds = Conexiones.Retorna_Datos(SQL);

            if (ds.Tables[0].Rows.Count > 0)
            {
                Conexiones.Ejecuta_Consulta("UPDATE trabajadores_obra set Telefono = " + Telefono + ", Nivel = "
                "" + Nivel + ", Contrato = " + Contrato + ", EPIs = " + EPIs + " , Maquinaria = " + Maquinaria + ", Apto = " + Apto +
                "," +
                );
            }
        }
    }
}

```

```

    " PRL = "" + PRL + "", Albañileria = "" + Albañileria + "", Hormigon = "" + Hormigon + "" , Grua = "" +
    Grua + "" , Plataformas = "" + Plataformas + "" , Libre = "" + Libre + "" , Obra_asignada = "" + Obra + """" +
    " where CODI = "" + CODI + """);

    MessageBox.Show("Registro modificado correctamente.");

    this.Close();
}
else
{
    Conexiones.Ejecuta_Consulta("INSERT INTO trabajadores_obra (CODI, Nombre, Apellidos, DNI,
    Telefono, Nivel, Contrato, EPIs, Maquinaria, Apto, PRL, Albañileria, Hormigon, Grua, Plataformas, Libre,
    Obra_asignada) " +
    "VALUES (" + CODI + ", " + Nombre + ", " + Apellidos + ", " + DNI + ", " + Telefono + ", " +
    Nivel + ", " + Contrato + ", " + EPIs + ", " + Maquinaria + ", " + Apto + ", " + PRL + ", " + Albañileria + ", " +
    Hormigon + ", " + Grua + ", " + Plataformas + ", " + Libre + ", " + Obra + ")");
}

MessageBox.Show("Registro guardado correctamente.");

    this.Close();
}
else
{
    MessageBox.Show("Debes insertar nombre, DNI, teléfono y si el trabajador no está asignado a una
obra (Libre).");
    return;
}
}
else
{
    if (Convert.ToString(CODI) != "" && Nombre != "" && DNI != "" && Telefono != "" && Libre != "")
    {
        Conexiones.Ejecuta_Consulta("INSERT INTO trabajadores_obra (CODI, Nombre, Apellidos, DNI,
        Telefono, Nivel, Contrato, EPIs, Maquinaria, Apto, PRL, Albañileria, Hormigon, Grua, Plataformas, Libre,
        Obra_asignada) " +
        "VALUES (" + CODI + ", " + Nombre + ", " + Apellidos + ", " + DNI + ", " + Telefono + ", " +
        Nivel + ", " + Contrato + ", " + EPIs + ", " + Maquinaria + ", " + Apto + ", " + PRL + ", " + Albañileria + ", " +
        Hormigon + ", " + Grua + ", " + Plataformas + ", " + Libre + ", " + Obra + ")");
    }

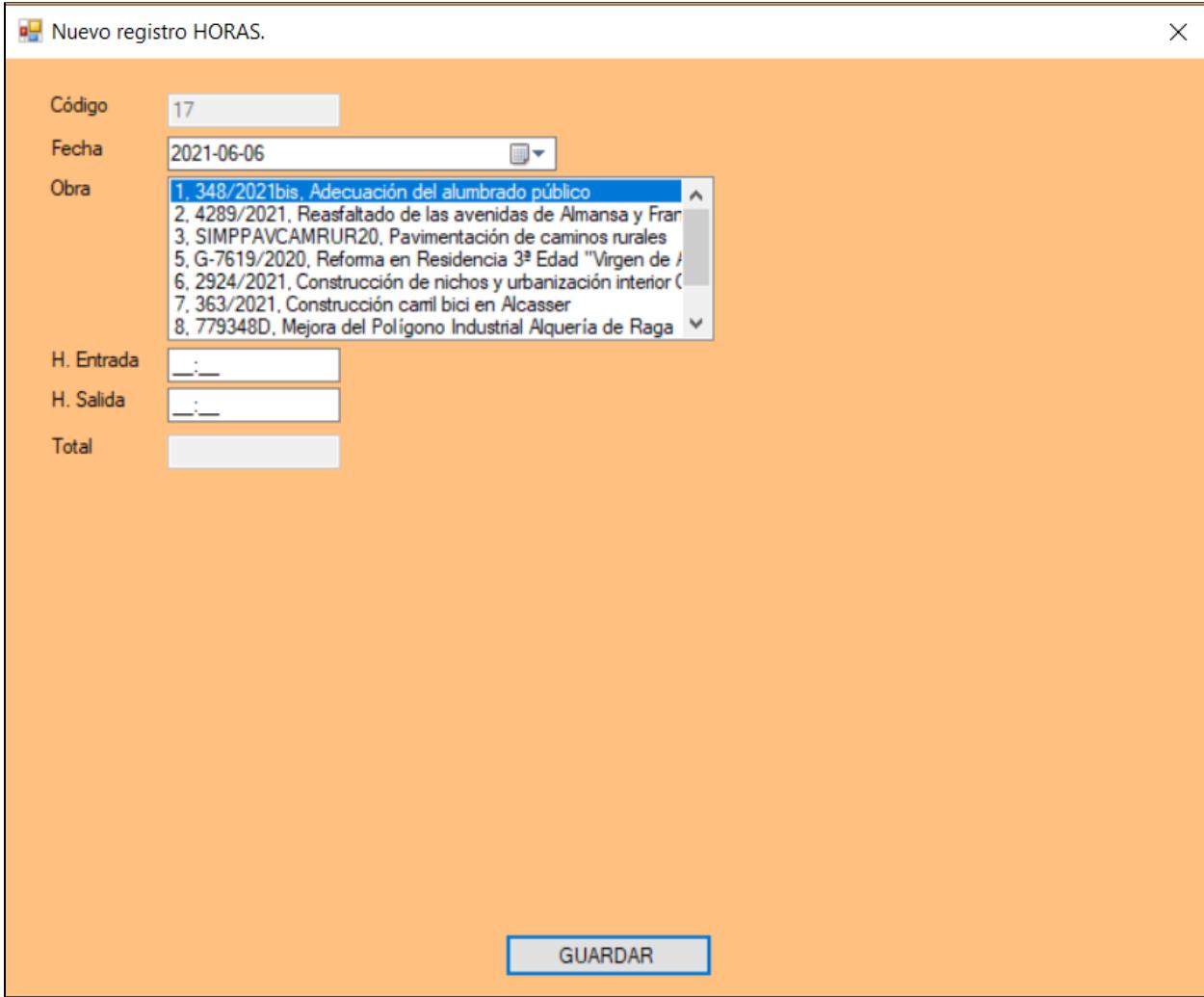
    MessageBox.Show("Registro guardado correctamente.");

    this.Close();
}
else
{
    MessageBox.Show("Debes insertar nombre, DNI, teléfono y si el trabajador no está asignado a una obra
(Libre).");
    return;
}
}
}

```

3.3. Problemas encontrados y sus soluciones

Entre los problemas surgidos se encuentra el cálculo de horas del formulario “Nuevo registro HORAS”.



The screenshot shows a Windows application window titled "Nuevo registro HORAS." The form contains the following fields:

- Código:** A text box containing the value "17".
- Fecha:** A date picker showing "2021-06-06".
- Obra:** A dropdown menu listing several construction projects:
 - 1. 348/2021bis. Adecuación del alumbrado público
 - 2. 4289/2021, Reasfaltado de las avenidas de Almansa y Fran
 - 3. SIMPPAVCAMRUR20, Pavimentación de caminos rurales
 - 4. G-7619/2020, Reforma en Residencia 3^a Edad "Virgen de A
 - 5. 2924/2021, Construcción de nichos y urbanización interior C
 - 6. 363/2021, Construcción carretera bici en Alcasser
 - 7. 779348D, Mejora del Polígono Industrial Alquería de Raga
 - 8. 779348D, Mejora del Polígono Industrial Alquería de Raga
- H. Entrada:** A text box containing ":".
- H. Salida:** A text box containing ":".
- Total:** A text box currently empty.
- GUARDAR:** A blue button at the bottom right of the form.

Primero se crearon dos “textBox” a través de los que el usuario podía introducir las horas y se recogían para realizar el cálculo. Se consideró deficiente este método, ya que cada usuario podría introducir los datos de una forma distinta, teniendo en cuenta un reloj de 12h o de 24h, introduciendo “.” para separar horas y minutos, o incluso algún usuario podría intentar insertar segundos, etc.

Por lo tanto, se decidió utilizar un elemento “maskedTextBox”, se tuvo que buscar información sobre la utilización de estos elementos. En el archivo de diseño, primero se inicializaron los componentes, según las siguientes líneas:

```
this.maskedTextBox1 = new System.Windows.Forms.MaskedTextBox();
this.maskedTextBox2 = new System.Windows.Forms.MaskedTextBox();
```

Tras esto, se definieron sus características, ambas con la misma estructura. Indicando la

localización, la máscara que debían mostrar, el tamaño y la acción asignada de validación.

```
// maskedTextBox1
//
this.maskedTextBox1.Location = new System.Drawing.Point(127, 206);
this.maskedTextBox1.Mask = "00:00";
this.maskedTextBox1.Name = "maskedTextBox1";
this.maskedTextBox1.Size = new System.Drawing.Size(132, 22);
this.maskedTextBox1.TabIndex = 31;
this.maskedTextBox1.Validating += new
System.ComponentModel.CancelEventHandler(this.maskedTextBox1_Validating);
```

El método para validar se insertó en el código con el objetivo de que el usuario no pudiese introducir valores mayores a 24 horas.

```
private void maskedTextBox1_Validating(object sender, CancelEventArgs e)
{
    Int32 horas = Int32.Parse(maskedTextBox1.Text.Substring(0, 2));
    if (horas > 24)
    {
        MessageBox.Show("No puede superar las 24hs");
        seleccionarhoras(maskedTextBox1);
        e.Cancel = true;
    }
}

private void seleccionarhoras(MaskedTextBox mkt)
{
    mkt.Focus();
    mkt.SelectionStart = 0;
    mkt.SelectionLength = 2;
}
```

Una vez comprobado que el usuario no podía insertar valores mayores a 24 horas, se escribió el código para el cálculo de las horas. Para ello, se crearon las variables necesarias, se indicó desde dónde se cogen los datos, en este caso, dos “maskedTextBox”. Se comprueba que ninguno es nulo y se pasan los valores a formato “TimeSpan” que representa un intervalo de tiempo. Por último, se realiza la resta de ambos valores “Substract” y se presenta en el “textBox” correspondiente.

```
private void Calcular_Total_Horas()
{
    string HEntrada = "";
    string HSalida = "";
    string THoraS = "";
    string THorasE = "";
    TimeSpan valor1;
    TimeSpan valor2;

    HEntrada = (maskedTextBox1.Text.ToString());
    HSalida = (maskedTextBox2.Text.ToString());

    if(maskedTextBox1.Text != "" && maskedTextBox2.Text != "")
    {
        THorasE = HEntrada.Substring(0, 2) + ":" + HEntrada.Substring(3, 2);
        THoraS = HSalida.Substring(0, 2) + ":" + HSalida.Substring(3, 2);
        valor1 = TimeSpan.Parse(THorasE);
        valor2 = TimeSpan.Parse(THoraS);
```

```

        TimeSpan diferencia = valor2.Subtract(valor1);
        textBox4.Text = Convert.ToString(diferencia);
    }
    else
    {
        MessageBox.Show("Debes introducir hora de entrada y de salida.");
    }
}

```

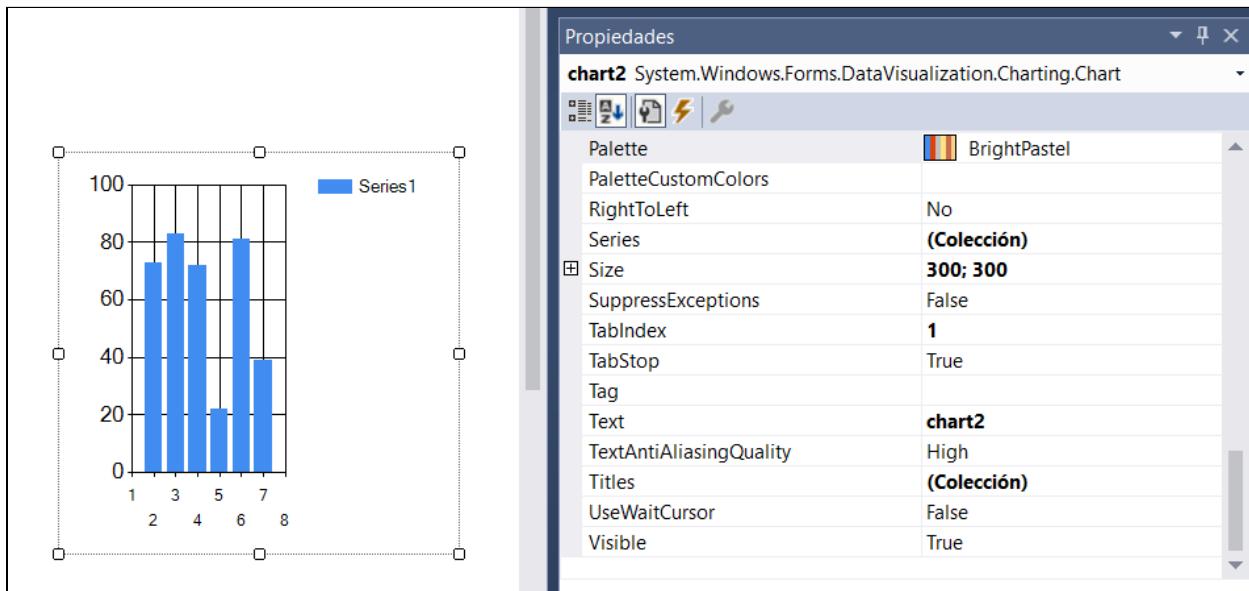
Tras hacer pruebas, se comprobó que existía un error, ya que se podían introducir valores mayores a 59 minutos, por lo tanto, se completo la validación de ambos campos, quedando de la siguiente manera:

```

private void maskedTextBox1_Validating(object sender, CancelEventArgs e)
{
    Int32 horas = Int32.Parse(maskedTextBox1.Text.Substring(0, 2));
    if (horas > 24)
    {
        MessageBox.Show("No puede superar las 24hs");
        seleccionarhoras(maskedTextBox1);
        e.Cancel = true;
    }
    Int32 minutos = Int32.Parse(maskedTextBox1.Text.Substring(3, 2));
    if (minutos > 59)
    {
        MessageBox.Show("No puede superar los 59min");
        seleccionarminutos(maskedTextBox1);
        e.Cancel = true;
    }
}

```

Otro de los problemas detectado está relacionado con la creación de las gráficas. Se utilizó un elemento “chart” de Microsoft Visual Studio. Al principio, los datos no se visualizaban, ya que se creaba el gráfico con los datos “por defecto”. Es decir, se visualizaba el título por defecto y las series/leyenda por defecto, la solución para esto fué eliminar estas propiedades que se muestran como “Colección”.



Posteriormente, se creó el método para cargar los datos del título y la leyenda, pero sólo se cargaba el primer dato que se obtenía de la consulta SQL. Si realizaba la consulta en la BBDD sí aparecían todos los datos según el sumatorio de gastos o de horas y la obra a la que pertenecían, pero en el gráfico sólo se visualizaba el primer dato y la obra correspondiente.

```
String SQL = "SELECT TITULO, SUM(Importe) AS Su FROM gastos GROUP BY TITULO";
```

Se decidió crear una lista en la que añadir los diferentes títulos obtenido de la consulta de datos, con esto se consiguió que aparecieran todos los títulos de las obras, pero en el momento que habían dos gastos u horas propios de una misma obra el programa se cerraba.

```
List<string> Obras_nombre = new List<string>();
private void Cargar_Chart()
{
    //String SQL = "SELECT YEAR(Fecha) AS yyyy, MONTHNAME(Fecha) as mm, SUM(Importe) AS Su FROM
gastos " +
    // "GROUP BY YEAR(Fecha), MONTH(Fecha)";

    String SQL = "SELECT TITULO, SUM(Importe) AS Su FROM gastos GROUP BY TITULO";
    DataSet ds = Conexiones.Retorna_Datos(SQL);

    DataTable firstTable = ds.Tables[0];

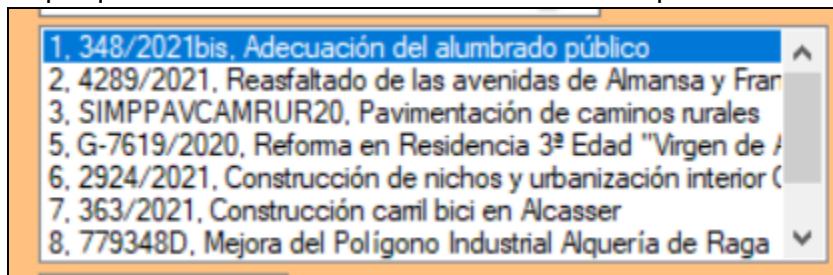
    chart1.Titles.Add("GASTOS REALIZADOS POR OBRAS");
    foreach (DataRow row in firstTable.Rows)
    {
        Series series = chart1.Series.Add(row["TITULO"].ToString());
        series.Points.Add(Convert.ToDouble(row["Su"].ToString()));
        series.Label = row["Su"].ToString();
    }
}
```

La solución fue introducir, antes de cargar el título del gráfico, la siguiente línea. Con esto se consigue que la leyenda se borre completamente antes de introducir los nuevos datos, por lo tanto, sólo se crea una vez aunque en la gráfica existan varios registros de una sola obra como

sumatorio.

```
chart1.Series.Clear();
```

El último problema que se va a comentar está relacionado con la creación del “listBox” que contiene el identificador, el expediente y el título de la obra. Es una información que se utiliza en varias pantallas para añadir horas a una obra en concreto o para asignar trabajadores a una obra, etc. En un principio se pensaba mostrar sólo el identificador, ya que suele ser la clave principal en la BBDD, pero, sin una definición se consideró que faltaba información y que el usuario no tendría porqué saber relacionar un ID interno con el expediente o título de la obra.



Aunque se intentó concatenar los datos de varias formas, la solución fue crear tres listas dentro del método “Rellenar_ListBox()”. Primero se crearon las variables y tras devolver los datos de la BBDD se realiza un bucle para añadir a cada lista los identificadores, los expedientes y los títulos. Por último, se creó una nueva lista “Lista_todo” donde se concatena ID, expediente y título de cada obra y se indica en el “listBox” correspondiente.

```
string ID = "";
string Exp = "";
string Ti = "";
List<string> Lista_ID_obra = new List<string>();
List<string> Lista_Exp_obra = new List<string>();
List<string> Lista_Ti_obra = new List<string>();

private void Rellenar_ListBox()
{
    //DataSet ds = Conexiones.Retorna_Datos("Select ID, Expediente, Titulo, Estado from obras where (Estado = 'PRESENTADA' or Estado = 'ADJUDICADA' or Estado = 'EN EJECUCIÓN' or Estado = 'FINALIZADA' or Estado = 'OTROS')");
    DataSet ds = Conexiones.Retorna_Datos("Select ID, Expediente, Titulo, Estado from obras");

    for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
    {
        ID = ds.Tables[0].Rows[i]["ID"].ToString();
        Exp = ds.Tables[0].Rows[i]["Expediente"].ToString();
        Ti = ds.Tables[0].Rows[i]["Titulo"].ToString();
        Lista_ID_obra.Add(ID);
        Lista_Exp_obra.Add(Exp);
        Lista_Ti_obra.Add(Ti);
    }

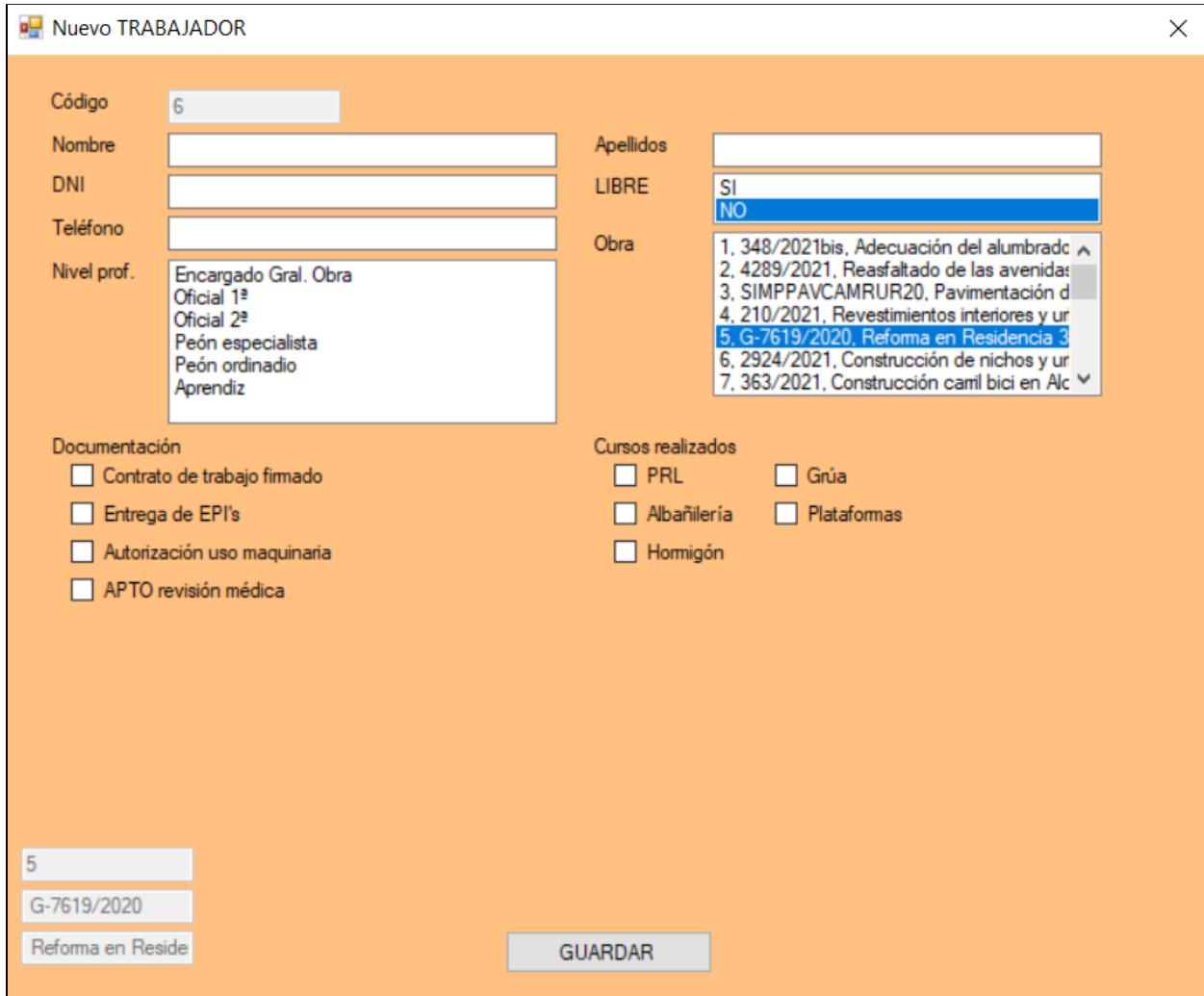
    List<string> Lista_todo = new List<string>();
    for (int i = 0; i < Lista_ID_obra.Count; i++)
    {
        string concatenar = Lista_ID_obra[i] + ", " + Lista_Exp_obra[i] + ", " + Lista_Ti_obra[i];
        Lista_todo.Add(concatenar);
    }
}
```

```

        Lista_todo.Add(concatenar);
    }
listBox3.DataSource = Lista_todo;
}

```

Por último, para guardar los datos del “listBox” a la BBDD se crearon tres campos en la parte inferior izquierda del formulario, aunque sólo están visibles para mostrar su funcionamiento en el formulario de “Nuevo TRABAJADOR”.



Nuevo TRABAJADOR

Código	6	Apellidos	
Nombre		LIBRE	<input checked="" type="checkbox"/> SI <input type="checkbox"/> NO
DNI		Obra	1, 348/2021bis, Adecuación del alumbrado 2, 4289/2021, Reasfaltado de las avenidas 3, SIMPPAVCAMRUR20, Pavimentación d... 4, 210/2021, Revestimientos interiores y ur... 5, G-7619/2020, Reforma en Residencia 3 6, 2924/2021, Construcción de nichos y ur... 7, 363/2021, Construcción carril bici en Alc...
Teléfono		Documentación	<input type="checkbox"/> Contrato de trabajo firmado <input type="checkbox"/> Entrega de EPI's <input type="checkbox"/> Autorización uso maquinaria <input type="checkbox"/> APTO revisión médica
Nivel prof.	Encargado Gral. Obra Oficial 1ª Oficial 2ª Peón especialista Peón ordinario Aprendiz	Cursos realizados	<input type="checkbox"/> PRL <input type="checkbox"/> Grúa <input type="checkbox"/> Albañilería <input type="checkbox"/> Plataformas <input type="checkbox"/> Hormigón
		5	
		G-7619/2020	
		Reforma en Reside	GUARDAR

El proceso es inverso al explicado anteriormente. En este caso también se crea una lista, se recoge cuál es el índice de la línea seleccionada y se recoge en un arreglo las palabras divididas por la “,”. Para cada una de estas palabras se comprueba que no existe en la “Lista_desconcatenar”, si es así se añade. Además, en los tres espacios inferiores se muestran los valores separados para comprobar que el funcionamiento es correcto.

```

List<string> Lista_desconcatenar = new List<string>();
private void listBox3_SelectedIndexChanged(object sender, EventArgs e)
{
    Lista_desconcatenar.Clear();
    string curItem = listBox3.SelectedItem.ToString();

```

```
int index = listBox3.FindString(curlItem);
string[] words = curlItem.Split(',');
foreach (var word in words)
{
    if (!Lista_desconectar.Contains(word))
    {
        Lista_desconectar.Add(word);
    }
}
for (int i = 0; i < 1; i++)
{
    textBox_ID.Text = Lista_desconectar[0];
    textBox_Exp.Text = Lista_desconectar[1];
    textBox_Tit.Text = Lista_desconectar[2];
}
```

4. AUTOEVALUACIÓN Y CONCLUSIONES

Con el desarrollo del proyecto creo que he podido poner en práctica los conocimientos adquiridos. En primer lugar, he sido capaz de reconocer las necesidades que se deben cubrir, para ello, he analizado y listado las tareas más comunes que se llevan a cabo a la hora de dirigir obras. A continuación, he esquematizado la información que se debe registrar para, posteriormente, crear la base de datos y cada una de las tablas que la componen. Seguidamente, he creado la interfaz gráfica desde la que el usuario se relacionará con los datos. Para finalizar, he ido adaptando el conjunto a las problemáticas que han ido surgiendo y a las opiniones e ideas de las personas a las que les he mostrado la aplicación.

Además, he podido afrontar ciertas dificultades o problemas, ya que he creado una base de datos desde cero. Para la aplicación he utilizado un lenguaje, C#, que no se ha tratado con amplitud durante la formación. Para el diseño he intentado que el programa fuera intuitivo y sencillo, aunque he tenido algunas dificultades para crear los gráficos o para actualizar los datos inmediatamente desde la aplicación a la base de datos.

Aunque considero que la aplicación es capaz de solucionar muchas de las tareas diarias realizadas por el departamento técnico de una empresa constructora, quedan pendientes una serie de funcionalidades para que el proyecto pueda ser desarrollado desde otros departamentos:

- En este caso no se ha realizado, ya que se considera que existen otros programas especializados, pero se podría generar una ventana para la creación de presupuestos, mediciones y certificaciones de obra.
- En el caso de que la empresa disponga de maquinaria propia se debería crear una agenda de maquinaria, para controlar dónde está cada una en cada momento, si está o no libre, control de mantenimiento, etc.
- Si se necesita controlar globalmente la empresa desde la misma aplicación, se deberían crear una serie de formularios para poder llevar el control financiero, contable y fiscal de la empresa.

En conclusión, opino que he afrontado correctamente el proyecto ya que he ido cumpliendo las pautas y objetivos previstos, además, he conseguido que la idea inicial se convierta en una aplicación útil e intuitiva. Aunque existen múltiples funcionalidades que se pueden ir agregando a la aplicación dependiendo de las futuras necesidades.

5. BIBLIOGRAFÍA

Aguilar Systems MX. (4 de mayo de 2021). *Como crear un repositorio y agregar un proyecto a GitHub desde Visual Studio Community 2019*. [Video]. Youtube.
<https://www.youtube.com/watch?v=560GOf3f6xk>

Compilar y generar en Visual Studio. Microsoft. (28 de mayo de 2021).
<https://docs.microsoft.com/es-es/visualstudio/ide/compiling-and-building-in-visual-studio?view=vs-2019>

Crear conexión a servidor en MySQL Workbench. Slideshare. (31 de marzo de 2021).
<https://es.slideshare.net/yayosyayo/crear-conexion-a-servidor-en-mysql-workbench>

Ejemplos de System.Data.Odbc OdbcCommand.ExecuteNonQuery en C# (CSharp). Hot Examples. (16 de abril de 2021).
<https://csharp.hotexamples.com/es/examples/System.Data.Odbc/OdbcCommand/ExecuteNonQuery/php-odbccommand-executenonquery-method-examples.html>

El Arte de la Programación. (4 de mayo de 2021). *Sincronizar repositorios desde Visual Studio a GitHub*. [Video]. Youtube. <https://www.youtube.com/watch?v=a0ID9rrQOWE&t=495s>

Esquema gráfico de la base MySQL con MySQL Workbench. Desarrolladoresweb.com. (15 de abril de 2021). <https://desarrolloweb.com/articulos/esquema-grafico-base-mysql-workbench.html>

Instalar Visual Studio. Microsoft. (5 de abril de 2021).
<https://docs.microsoft.com/es-es/visualstudio/install/install-visual-studio?view=vs-2019>

OdbcCommand Clase. Microsoft. (15 de abril de 2021).
<https://docs.microsoft.com/es-es/dotnet/api/system.data.odbc.odbccommand?view=dotnet-plat-ext-5.0>

OdbcConnection Clase. Microsoft. (15 de abril de 2021).
<https://docs.microsoft.com/es-es/dotnet/api/system.data.odbc.odbccnection?view=dotnet-plat-ext-5.0>

OdbcDataAdapter Clase. Microsoft. (16 de abril de 2021).
<https://docs.microsoft.com/es-es/dotnet/api/system.data.odbc.odbcdataladapter?view=dotnet-plat-ext-5.0>

Primer vistazo al depurador de Visual Studio. Microsoft. (12 de abril de 2021).
<https://docs.microsoft.com/es-es/visualstudio/debugger/debugger-feature-tour?view=vs-2019>

Tecnología Binaria. (25 de mayo de 2021). *Cómo HACER un GRÁFICO en Visual Studio C# | Control CHART* [Video]. Youtube. <https://www.youtube.com/watch?v=X64uVTxRe6M&t=454s>

TimeSpan Estructura. Microsoft. (18 de mayo de 2021).
<https://docs.microsoft.com/es-es/dotnet/api/system.timespan?view=net-5.0>

Tutorial: Creación de una aplicación de consola de C# sencilla en Visual Studio. *Microsoft*. (6 de abril de 2021).

<https://docs.microsoft.com/es-es/visualstudio/get-started/csharp/tutorial-console?view=vs-2019>

¿Cómo conectar MySQL a Workbench?. *HostGator*. (31 de marzo de 2021).

<https://soporte-latam.hostgator.com/hc/es-419/articles/360004397732--C%C3%B3mo-conectar-MySQL-a-Workbench->

6. ANEXOS

6.1. Manual del usuario

Dacobrick se compone por una serie de pestañas que van apareciendo/desapareciendo según las indicaciones del usuario a través de botones.

De manera general, los campos de texto con fondo gris están bloqueados y no son modificables, los de fondo blanco son los que requieren la introducción de texto.

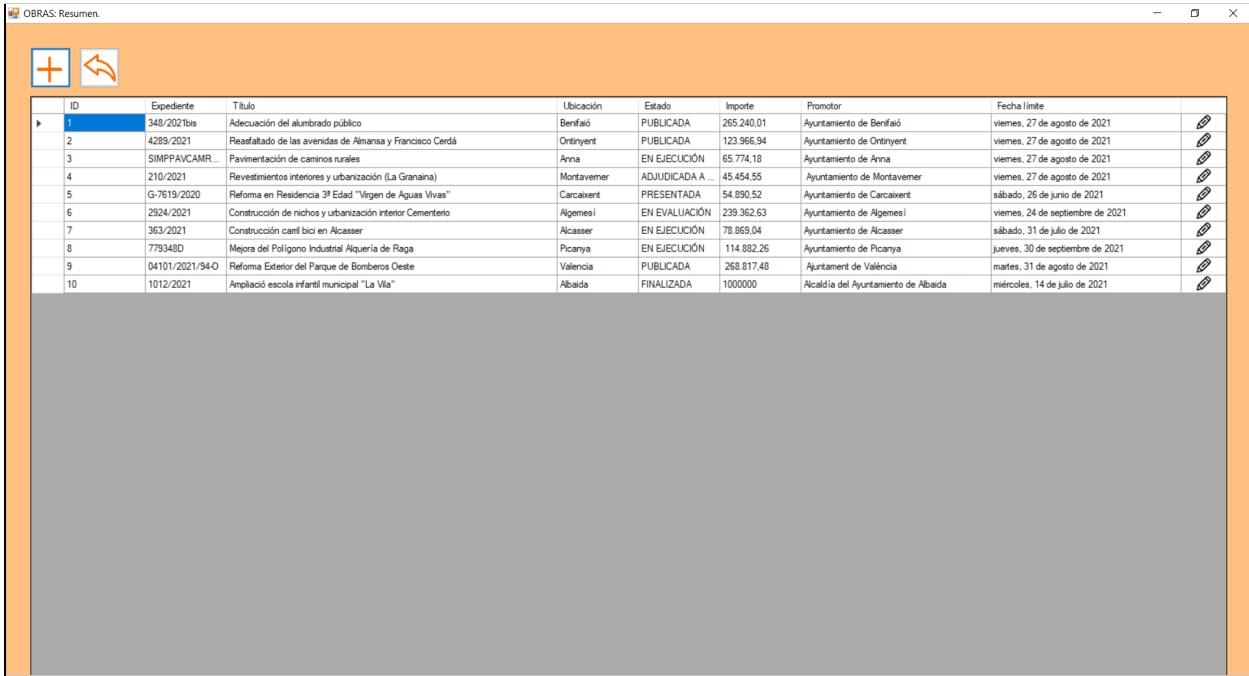
Al iniciar la aplicación, aparece la pantalla de inicio que muestra el logotipo y el eslogan de Dacobrick. Además, contiene un botón “ENTRAR” que nos permite el acceso al menú principal.



En el menú principal existen 4 opciones: Obras, personal, planificación y otros.



Desde la opción de obras, se accede al listado de construcciones existentes. En el caso de que todavía no exista ninguna, este listado aparecerá vacío. El primer botón en la parte superior izquierda con el símbolo “+” permite crear una nueva obra. El botón de la fecha permite volver a la pantalla de atrás, es decir, al menú principal.



The screenshot shows a Windows application window titled "OBRA'S: Resumen". At the top left are two buttons: a blue square with a white plus sign (+) and a red square with a white arrow pointing left. The main area is a table with the following data:

ID	Expediente	Título	Ubicación	Estado	Importe	Promotor	Fecha límite
1	348/2021bis	Adecuación del alumbrado público	Benifaió	PUBLICADA	265.240,01	Ayuntamiento de Benifaió	viernes, 27 de agosto de 2021
2	4289/2021	Resalfatado de las avenidas de Almània y Francisco Cerdà	Ontinyent	PUBLICADA	123.966,94	Ayuntamiento de Ontinyent	viernes, 27 de agosto de 2021
3	SIMPACAMR.	Pavimentación de caminos rurales	Anna	EN EJECUCIÓN	65.774,18	Ayuntamiento de Anna	viernes, 27 de agosto de 2021
4	210/2021	Revestimientos interiores y urbanización (La Granana)	Montavemar	ADJUDICADA A	45.454,55	Ayuntamiento de Montavemar	viernes, 27 de agosto de 2021
5	G-619/2020	Reforma en Residencia 3ª Edad "Virgen de Aguas Vivas"	Carcassent	PRESENTADA	54.990,52	Ayuntamiento de Carcassent	sábado, 26 de junio de 2021
6	2924/2021	Construcción de nichos y urbanización interior Cementerio	Algemesí	EN EVALUACIÓN	239.362,63	Ayuntamiento de Algemesí	viernes, 24 de septiembre de 2021
7	363/2021	Construcción canf bici en Alcasser	Alcasser	EN EJECUCIÓN	78.869,04	Ayuntamiento de Alcasser	sábado, 31 de julio de 2021
8	779348D	Mejora del Polígono Industrial Alqueria de Raga	Picanya	EN EJECUCIÓN	114.882,26	Ayuntamiento de Picanya	jueves, 30 de septiembre de 2021
9	04101/2021-94-O	Reforma Exterior del Parque de Bomberos Oeste	València	PUBLICADA	268.817,48	Ayuntamiento de València	martes, 31 de agosto de 2021
10	1012/2021	Ampliació escola infantil municipal "La Vila"	Albaida	FINALIZADA	1000000	Alcaldía del Ayuntamiento de Albaida	miércoles, 14 de julio de 2021

Si se pulsa el botón “+”, se accede a la siguiente ventana.

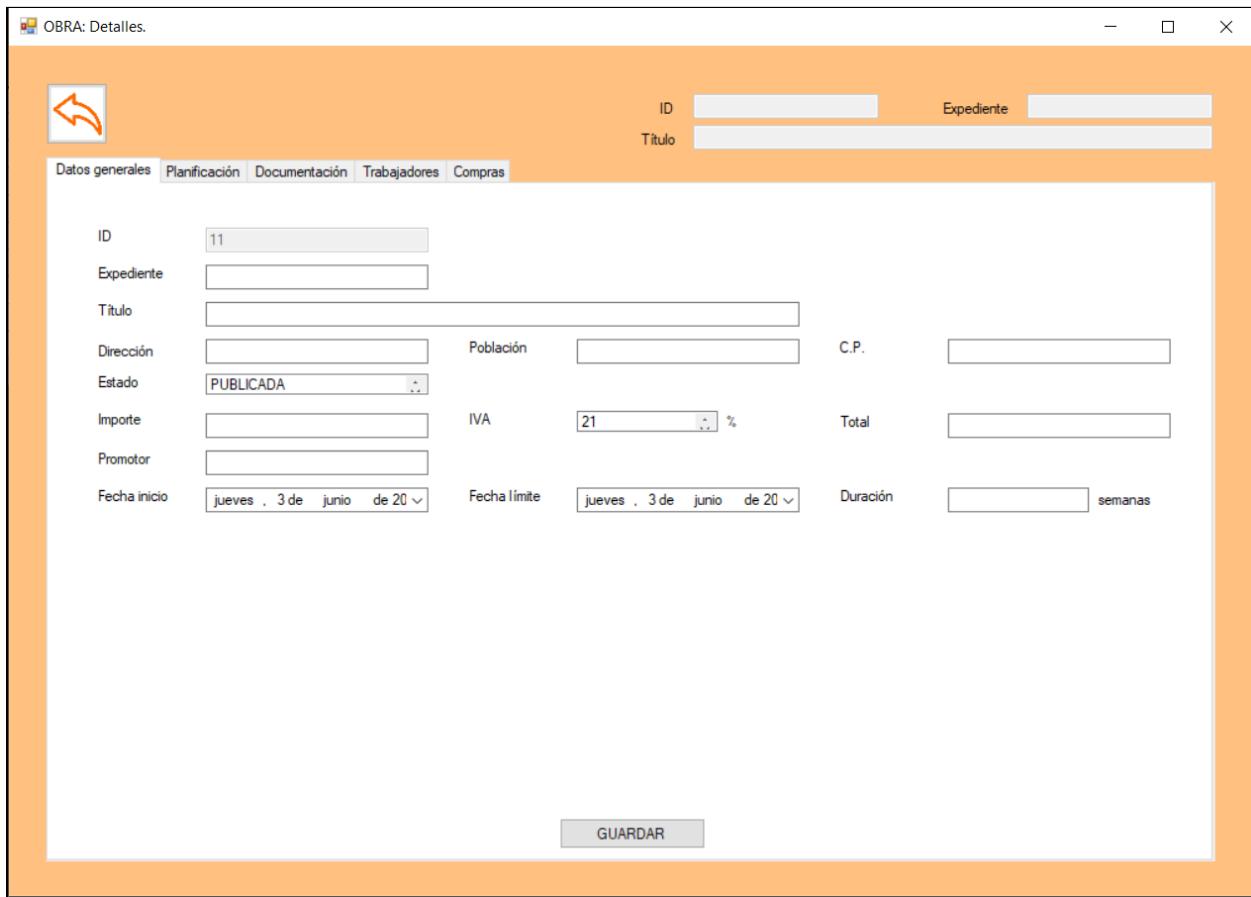
En la parte superior izquierda se encuentra el botón para volver atrás y, en la parte superior derecha existen una serie de campos que se rellenan automáticamente, sirven de guía para saber en qué obra nos encontramos, su identificador, su expediente y su título.

Esta pantalla se compone de varias pestañas. En la primera pestaña se dan de alta los datos generales de la obra, el ID es el identificador de cada una de ellas, se compone de un número único por lo que se calcula automáticamente y no es modificable. El resto de campos son:

- Expediente, para guardar la referencia externa de la obra.
- Título, el nombre de la obra.
- Dirección, población y C.P., para indicar dónde se encuentra la edificación.
- Estado, este campo puede ser modificado con posterioridad y los estados existentes son: publicada, presentada, en evaluación, adjudicada, en ejecución, adjudicada a otros, anulada, finalizada y otros.
- Importe, I.V.A. y total, se completarán con estos datos. El I.V.A. puede ser del 21% o del 10%. Estos datos podrán ser modificados con posterioridad.
- Promotor, dónde se indicará la persona o entidad que impulsa la construcción.
- Fecha de inicio, fecha límite y duración. Cabe destacar que la duración se debe señalar en semanas. Estos datos además son modificables.

Se debe pulsar “GUARDAR” antes de continuar a la siguiente pestaña.

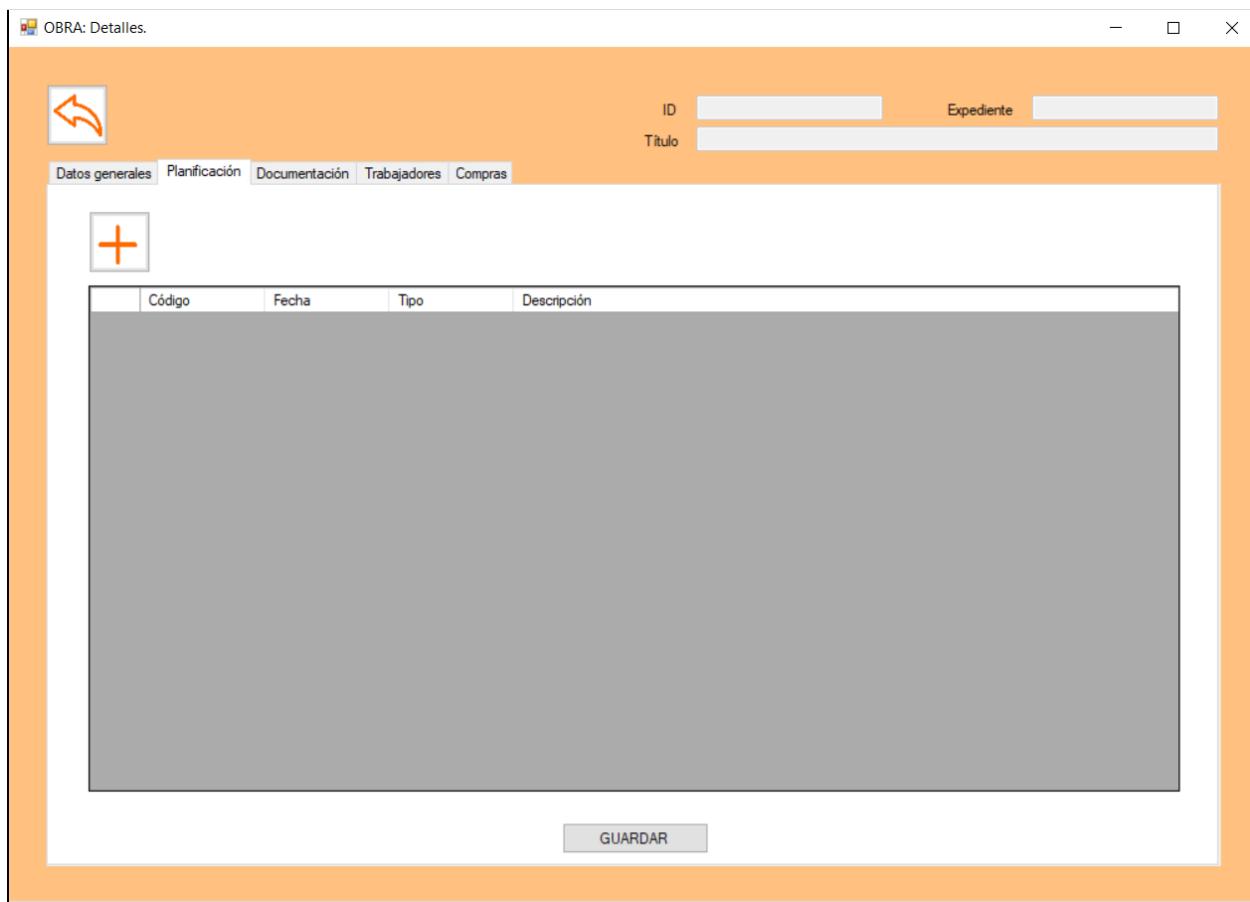
OBRA: Detalles.



ID	11	Expediente	
Expediente		Título	
Dirección		Población	
Estado	PUBLICADA	C.P.	
Importe		IVA	21 %
Promotor		Total	
Fecha inicio	jueves , 3 de junio de 2019	Fecha límite	jueves , 3 de junio de 2019
		Duración	semanas

GUARDAR

La siguiente pestaña, planificación, permite visualizar los eventos programados para la obra en la que nos encontramos. Para crear nuevos eventos se deberá pulsar el botón “+”.



Al pulsar este botón aparecerá una ventana para crear una nueva planificación. Se debe tener en cuenta que no son modificables, es decir, una vez creadas no se pueden ni modificar ni eliminar.

El identificador viene dado por defecto, se deberá introducir la fecha utilizando el botón desplegable de la derecha del campo. A continuación, se seleccionará a qué fase de la obra se asigna el evento: Actuaciones previas, demoliciones, estructura, instalaciones u otros. Por último, se podrá incluir una descripción más concreta. Antes de salir, se deberá pulsar "GUARDAR".

Nueva PLANIFICACIÓN

Código	10
Fecha	03-06-2021
Tipo	<input type="checkbox"/> Actuaciones previas <input type="checkbox"/> Demoliciones <input type="checkbox"/> Acondicionamiento del terreno <input type="checkbox"/> Cimentaciones <input type="checkbox"/> Estructuras
Descripción	<input type="text"/>

GUARDAR

La tercera pestaña se utiliza para marcar la documentación que se ha ido recogiendo, está separada en tres bloques: Inicio de obra, durante su desarrollo y a su fin. Aún así, es independiente se puede marcar documentación de final de obra sin tener toda la documentación anterior marcada, esto es porque no todas las construcciones son iguales y no en todas se necesita la misma documentación.

OBRA: Detalles.

ID: Expediente:
Título:



[Datos generales](#) [Planificación](#) [Documentación](#) **Trabajadores** [Compras](#)

INICIO de obra

Contrato Firmado Libro de Órdenes Licencia/Declaración Apertura Centro Plan Gestión Residuos Proyecto Grúa
 Proyecto Completo Aviso Previo Plan Seguridad Salud Libro de Incidencias Libro Subcontratación Acta de Replanteo
 Estudio Geotécnico Nombram. prevención Libro de Calidades

DURANTE la obra

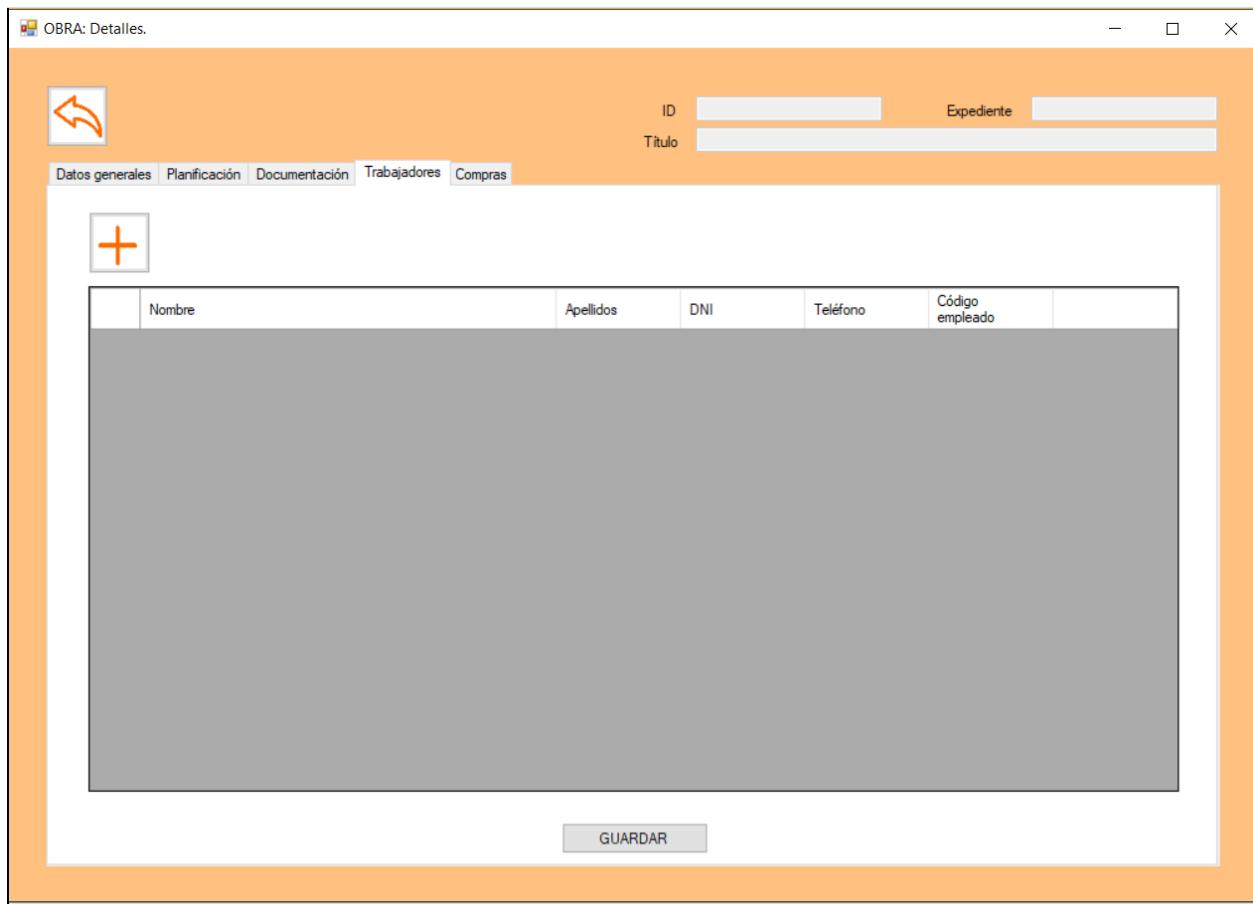
Doc. Cimentación Doc. Fachadas Doc. Carpintería Doc. Impermeabiliz. Doc. Revestimientos
 Doc. Estructura Doc. Particiones Doc. Aislamientos Doc. Instalaciones Doc. Equipamiento

FIN de obra

Liquidación Final Acta de Recepción
 Certificado Final Cert. Eficiencia

GUARDAR

En la cuarta pestaña, trabajadores, se mostrará un listado de los empleados asociados a la obra.



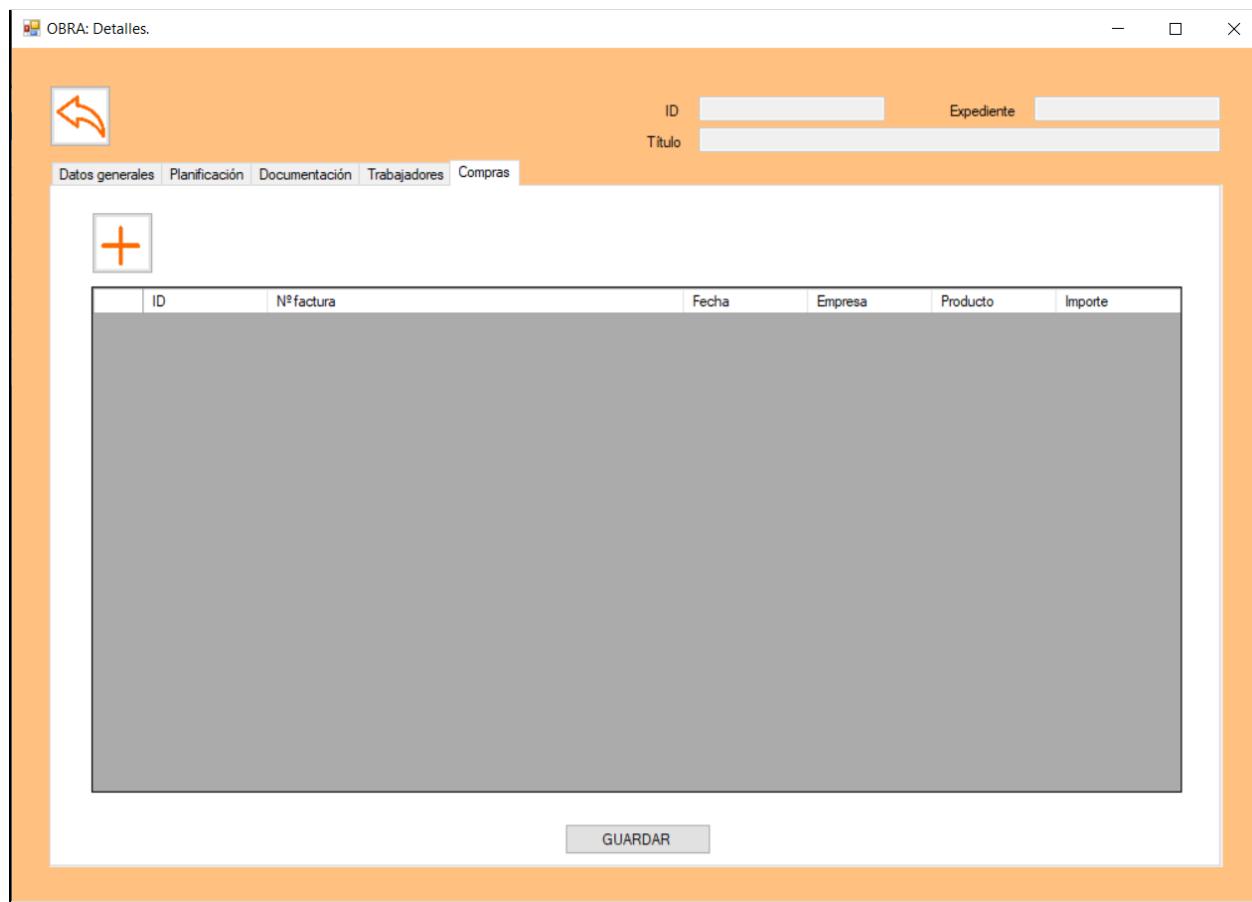
Pulsando sobre el botón “+” de esta pestaña, se abrirá una nueva ventana. En esta nueva ventana, aparecerá un listado de los trabajadores existentes en la base de datos que estén libres, es decir, que no tengan asociada ninguna obra.



Nuevo TRABAJADOR en obra

Código empleado	Nombre	Apellidos	Seleccionar

La última pestaña es para ver las compras existentes en la obra y, con el botón “+” se crearán nuevas compras.



Cada nueva compra deberá incluir el nº de factura, la fecha, la empresa en la que se ha realizado la compra, el producto y el importe. Se deberá pulsar “GUARDAR” para que quede grabado en la base de datos.

Nueva COMPRA.

Código	8
Nº. factura	<input type="text"/>
Fecha	2021-06-03 <input type="button" value=""/>
Empresa	<input type="text"/>
Producto	<input type="text"/>
Importe	<input type="text"/>

GUARDAR

Nuevamente, desde la pantalla de “OBRAS: Resumen”, se puede también acceder a cada construcción, ya sea para modificarla o para revisar los registros hechos. Para ello, desde la lista de obras, se pulsará el botón del lápiz , al final de cada línea.

OBRAS: Resumen.

ID	Expediente	Título	Ubicación	Estado	Importe	Promotor	Fecha límite
1	349/2021bis	Adecuación del alumbrado público	Benifaió	PUBLICADA	265.240,01	Ayuntamiento de Benifaió	viernes, 27 de agosto de 2021
2	4289/2021	Reasfaltado de las avenidas de Almansa y Francisco Cerdá	Ontinyent	PUBLICADA	123.966,94	Ayuntamiento de Ontinyent	viernes, 27 de agosto de 2021
3	SIMPRAVACAMR...	Pavimentación de caminos rurales	Anna	EN EJECUCIÓN	65.774,18	Ayuntamiento de Anna	viernes, 27 de agosto de 2021
4	210/2021	Revestimientos interiores y urbanización (La Granaina)	Montavemar	ADJUDICADA A ...	45.454,55	Ayuntamiento de Montavemar	viernes, 27 de agosto de 2021
5	G-7619/2020	Reforma en Residencia 3ª Edad "Virgen de Aguas Vivas"	Carcaxent	PRESENTADA	54.890,52	Ayuntamiento de Carcaxent	sábado, 26 de junio de 2021
6	2924/2021	Construcción de nichos y urbanización interior Cementerio	Algemesí	EN EVALUACIÓN	239.362,63	Ayuntamiento de Algemesí	viernes, 24 de septiembre de 2021
7	363/2021	Construcción camí bici en Alcàsser	Alcàsser	EN EJECUCIÓN	78.869,04	Ayuntamiento de Alcàsser	sábado, 31 de julio de 2021
8	77934BD	Mejora del Polígono Industrial Alquería de Raga	Picanya	EN EJECUCIÓN	114.882,26	Ayuntamiento de Picanya	jueves, 30 de septiembre de 2021
9	04101/2021/94-O	Reforma Exterior del Parque de Bomberos Oeste	València	PUBLICADA	268.817,48	Ajuntament de València	martes, 31 de agosto de 2021
10	1012/2021	Ampliació escola infantil municipal "La Vila"	Albaida	FINALIZADA	1000000	Alcaldía del Ayuntamiento de Albaida	miércoles, 14 de julio de 2021

De esta forma, se accede a la ventana de cada edificación, con las pestañas ya comentadas anteriormente.

En este caso, desde la pestaña de datos generales sólo se nos permitirá cambiar el estado, los importes, las fechas y la duración de la obra. Se debe pulsar “GUARDAR” para registrar los datos.

OBRA: Detalles.

	ID: 7	Expediente: 363/2021
	Título: Construcción camí bici en Alcasser	
Datos generales Planificación Documentación Trabajadores Compras		
ID: 7	Población: Alcasser	C.P.: 46290
Expediente: 363/2021	Estado: EN EJECUCIÓN	Importe: 78.869,04 IVA: 21 % Total: 95.431,54
Dirección: Plaza del Castell, 1		
Promotor: Ayuntamiento de Alcasser		
Fecha inicio: lunes , 31 de mayo de 20	Fecha límite: sábado , 31 de julio de 20	Duración: 8 semanas
<input type="button" value="GUARDAR"/>		

En la pestaña de planificación se podrán revisar los eventos programados y se podrán crear nuevos:

	ID: 7	Expediente: 363/2021																
	Título: Construcción camí bici en Alcasser																	
Datos generales Planificación Documentación Trabajadores Compras																		
	<table border="1"> <thead> <tr> <th>Código</th> <th>Fecha</th> <th>Tipo</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>10</td> <td>01-06-2021</td> <td>Actuaciones prev...</td> <td>Vista a obra, inicio de obra, firma acta de replanteo.</td> </tr> <tr> <td>11</td> <td>03-06-2021</td> <td>Demoliciones</td> <td>Recepción de la maquinaria para demoliciones.</td> </tr> <tr> <td>12</td> <td>07-06-2021</td> <td>OTROS</td> <td>Cita en el Ayto., revisión del programa de trabajo.</td> </tr> </tbody> </table>		Código	Fecha	Tipo	Descripción	10	01-06-2021	Actuaciones prev...	Vista a obra, inicio de obra, firma acta de replanteo.	11	03-06-2021	Demoliciones	Recepción de la maquinaria para demoliciones.	12	07-06-2021	OTROS	Cita en el Ayto., revisión del programa de trabajo.
Código	Fecha	Tipo	Descripción															
10	01-06-2021	Actuaciones prev...	Vista a obra, inicio de obra, firma acta de replanteo.															
11	03-06-2021	Demoliciones	Recepción de la maquinaria para demoliciones.															
12	07-06-2021	OTROS	Cita en el Ayto., revisión del programa de trabajo.															

En documentación podremos revisar la existente o registrar la nueva:

OBRA: Detalles.

ID: 7 Expediente: 363/2021

Título: Construcción camí bici en Alcasser

Datos generales

INICIO de obra

Contrato Firmado Libro de Órdenes Licencia/Declaración Apertura Centro Plan Gestión Residuos Proyecto Grúa
 Proyecto Completo Aviso Previo Plan Seguridad Salud Libro de Incidencias Libro Subcontratación Acta de Replanteo
 Estudio Geotécnico Nombram. prevención Libro de Calidades

DURANTE la obra

Doc. Cimentación Doc. Fachadas Doc. Carpintería Doc. Impermeabiliz. Doc. Revestimientos
 Doc. Estructura Doc. Particiones Doc. Aislamientos Doc. Instalaciones Doc. Equipamiento

FIN de obra

Liquidación Final Acta de Recepción
 Certificado Final Cert. Eficiencia

GUARDAR

En la pestaña de trabajadores podremos ver los que están adheridos a la obra. Se podrán agregar nuevos, ya explicado anteriormente y, también se podrán eliminar. Para suprimirlos, se

deberá pulsar la papelera  situada al final en cada línea del listado. Nos aparecerá un mensaje para confirmar si queremos eliminar o no al obrero de dicha obra, en caso afirmativo el trabajador quedará “libre” para poder añadirlo a otra.

OBRA: Detalles.

ID: 7 Expediente: 363/2021

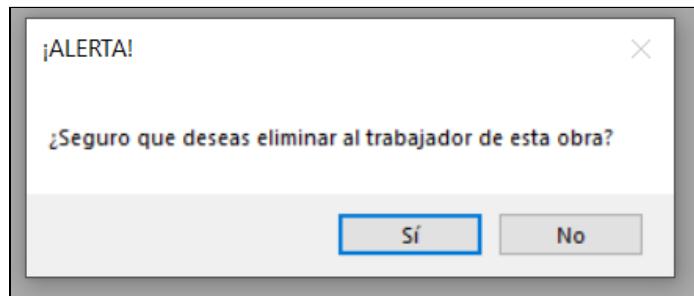
Título: Construcción camí bici en Alcasser

Datos generales

Trabajadores

+ Agregar

	Nombre	Apellidos	DNI	Teléfono	Código empleado	
▶	Jose	Moreno Navarro	11111111B	611111111	2	
	Carmen	Gil Ramos	22222222C	622222222	3	

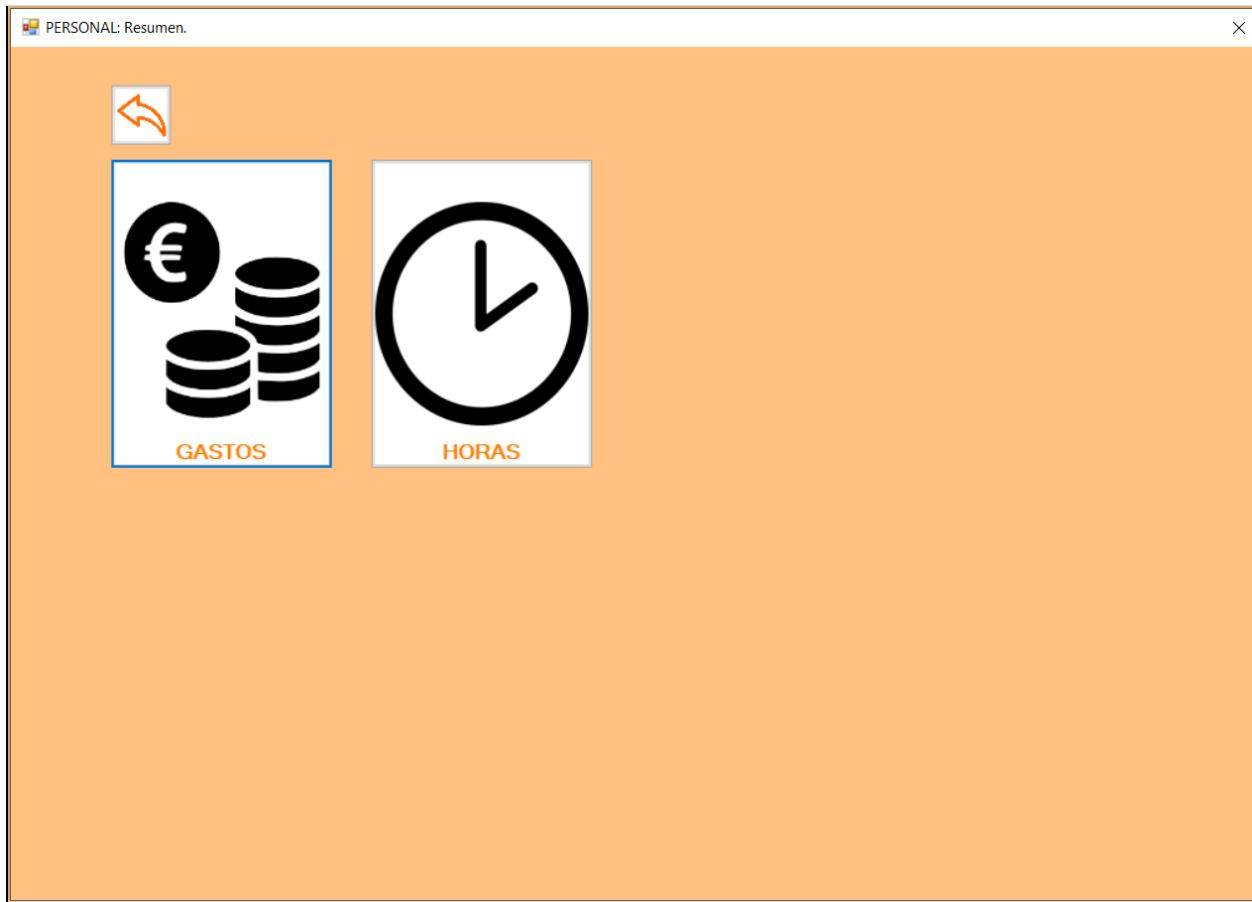


Por último, desde la pestaña compras se podrán revisar las ya realizadas o agregar nuevas:

The screenshot shows a software interface with an orange header bar. On the left, there's a back arrow icon. In the center, the ID is set to 7 and the Expediente (File) number is 363/2021. Below this, the Title is "Construcción carretera bici en Alcàsser". Underneath the header, there are several tabs: Datos generales, Planificación, Documentación, Trabajadores, and Compras. The Compras tab is currently selected, indicated by a blue border. To the right of the tabs, there's a large button with a plus sign (+). Below these elements is a table showing purchase details:

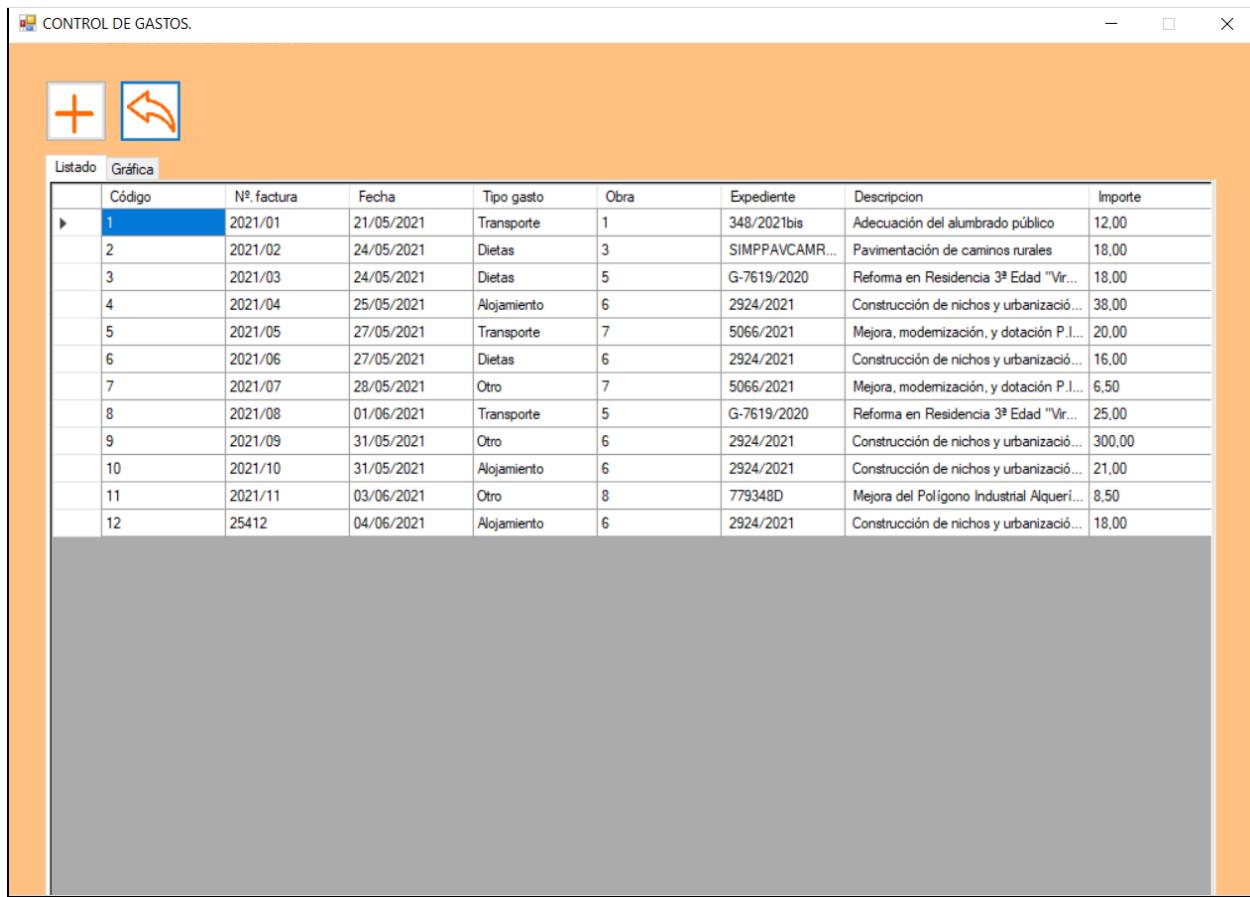
	ID	Nº factura	Fecha	Empresa	Producto	Importe
▶	8	2021/08	2021-06-01	Ayto. de Alcàsser	Pago tasas	22,00
	9	2021/09	2021-06-03	Ferretería San Ju...	Guantos talla M x 2	8,50

Volviendo al menú principal, si se accede a “PERSONAL”, se abrirá el siguiente submenu, con las opciones de “GASTOS” y “HORAS”.



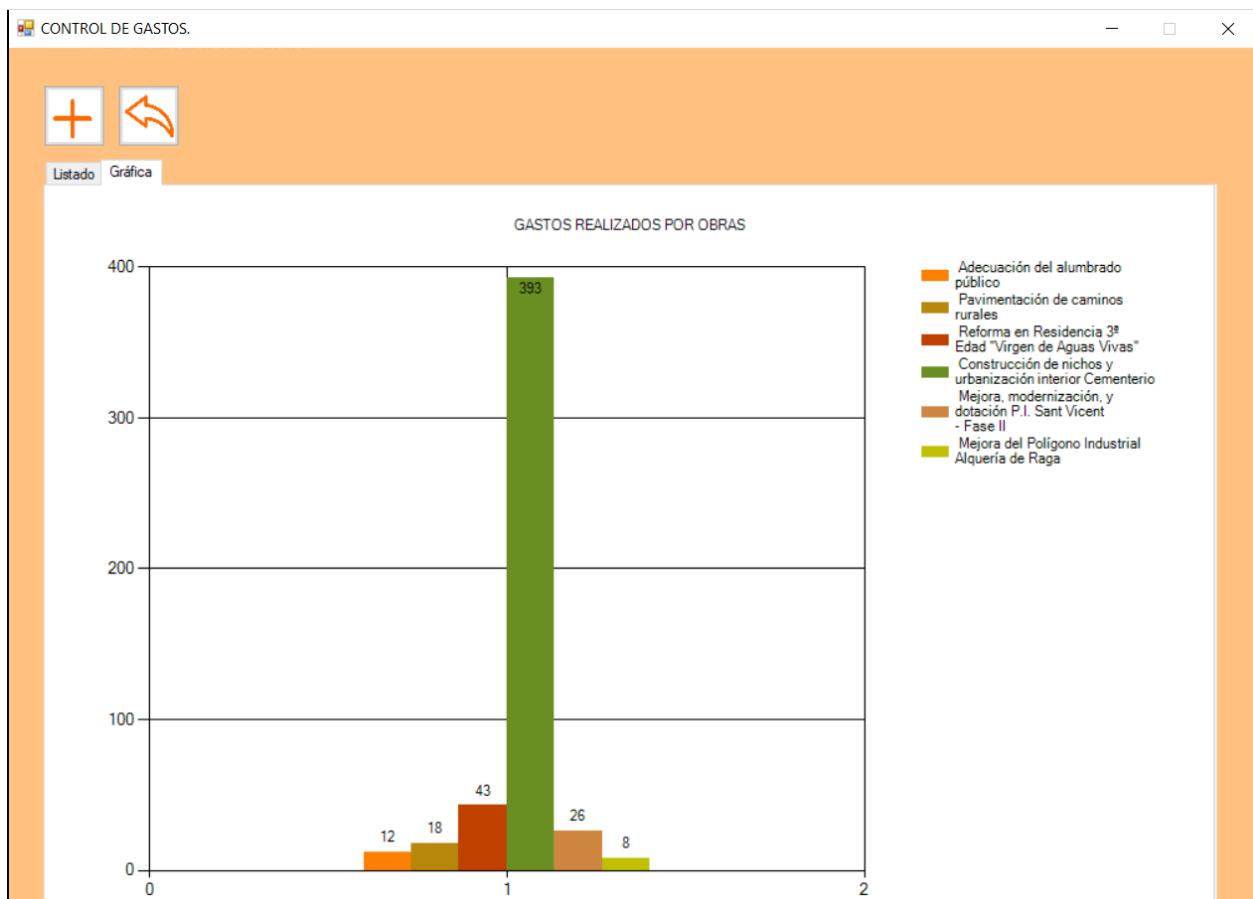
Dentro de cada una de estas opciones encontramos formularios muy similares, ambos compuestos por un listado y por una gráfica. Desde el control de gastos podremos ver el listado de los realizados según un identificador único, el nº de factura, la fecha, el tipo de desembolso, la obra (id, expediente y título) y el importe.

CONTROL DE GASTOS.



Código	Nº. factura	Fecha	Tipo gasto	Obra	Expediente	Descripción	Importe
1	2021/01	21/05/2021	Transporte	1	348/2021bis	Adecuación del alumbrado público	12,00
2	2021/02	24/05/2021	Dietas	3	SIMPPAVCAMR...	Pavimentación de caminos rurales	18,00
3	2021/03	24/05/2021	Dietas	5	G-7619/2020	Reforma en Residencia 3ª Edad "Vir...	18,00
4	2021/04	25/05/2021	Alojamiento	6	2924/2021	Construcción de nichos y urbanizació...	38,00
5	2021/05	27/05/2021	Transporte	7	5066/2021	Mejora, modernización, y dotación P.I...	20,00
6	2021/06	27/05/2021	Dietas	6	2924/2021	Construcción de nichos y urbanizació...	16,00
7	2021/07	28/05/2021	Otro	7	5066/2021	Mejora, modernización, y dotación P.I...	6,50
8	2021/08	01/06/2021	Transporte	5	G-7619/2020	Reforma en Residencia 3ª Edad "Vir...	25,00
9	2021/09	31/05/2021	Otro	6	2924/2021	Construcción de nichos y urbanizació...	300,00
10	2021/10	31/05/2021	Alojamiento	6	2924/2021	Construcción de nichos y urbanizació...	21,00
11	2021/11	03/06/2021	Otro	8	779348D	Mejora del Polígono Industrial Alqueri...	8,50
12	25412	04/06/2021	Alojamiento	6	2924/2021	Construcción de nichos y urbanizació...	18,00

Los mismos gastos podrán visualizarse en la gráfica de la segunda pestaña, se muestran agrupados por obras.



Desde el botón “+” se podrán crear nuevos gastos completando los datos solicitados en el formulario “Nuevo GASTO”. Se deberá rellenar el nº de factura, se seleccionará la fecha desde el desplegable, se elegirá un tipo de gasto y la obra y, por último se incluirá el importe. Se deberá guardar antes de volver a la ventana anterior.



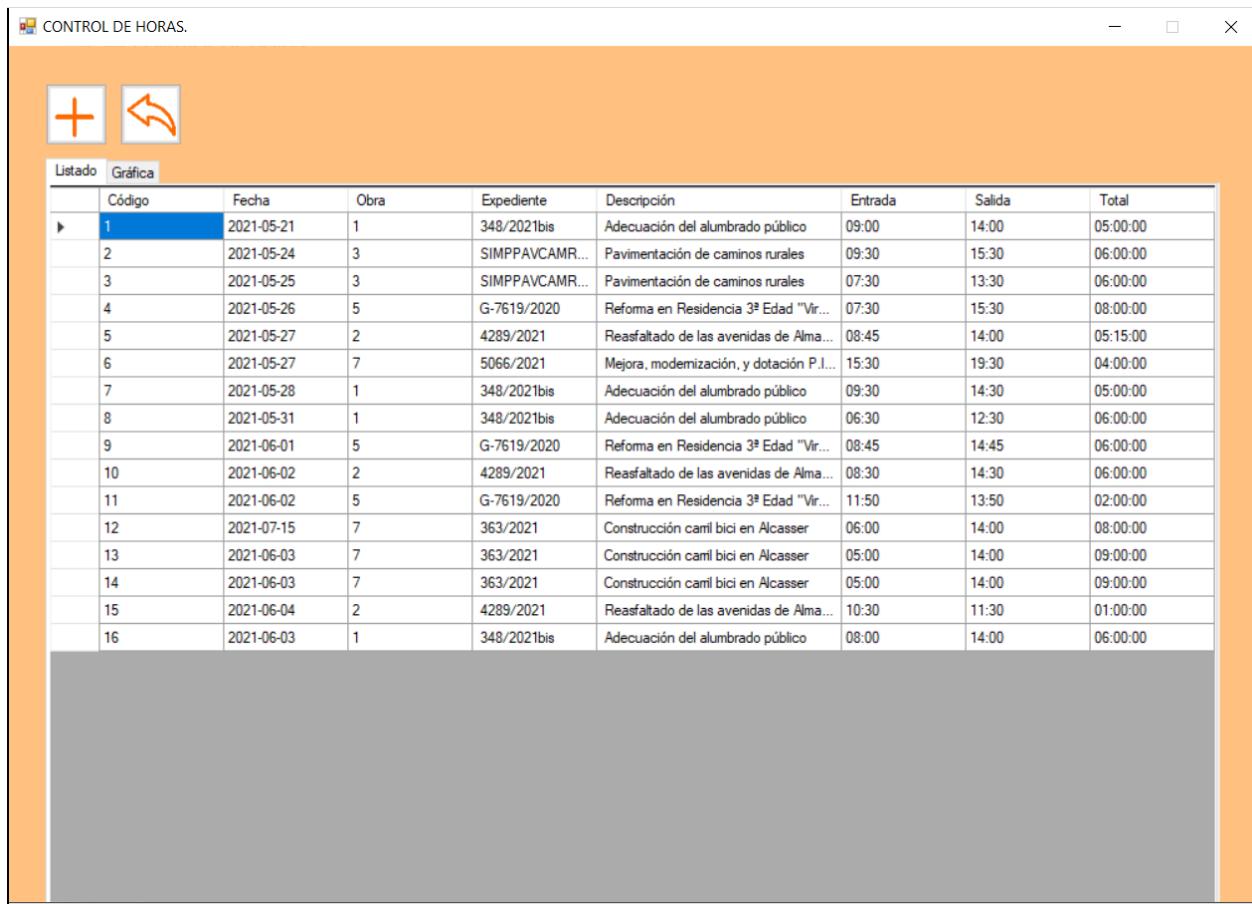
Nuevo GASTO.

Código	13
Nº. factura	<input type="text"/>
Fecha	2021-06-03
Tipo	Transporte Dietas Alojamiento Otro
Obra	1. 348/2021bis. Adecuación del alumbrado público 2. 4289/2021, Reasfaltado de las avenidas de Almansa y Fran 3. SIMPPAVCAMRUR20, Pavimentación de caminos rurales 5. G-7619/2020, Reforma en Residencia 3 ^a Edad "Virgen de / 6. 2924/2021, Construcción de nichos y urbanización interior C 7. 363/2021, Construcción caril bici en Alcasser 8. 779348D, Mejora del Polígono Industrial Alqueria de Raga
Importe	<input type="text"/>

GUARDAR

El submenú “HORAS” también muestra un formulario con listado y gráfica, muy similar al anterior.

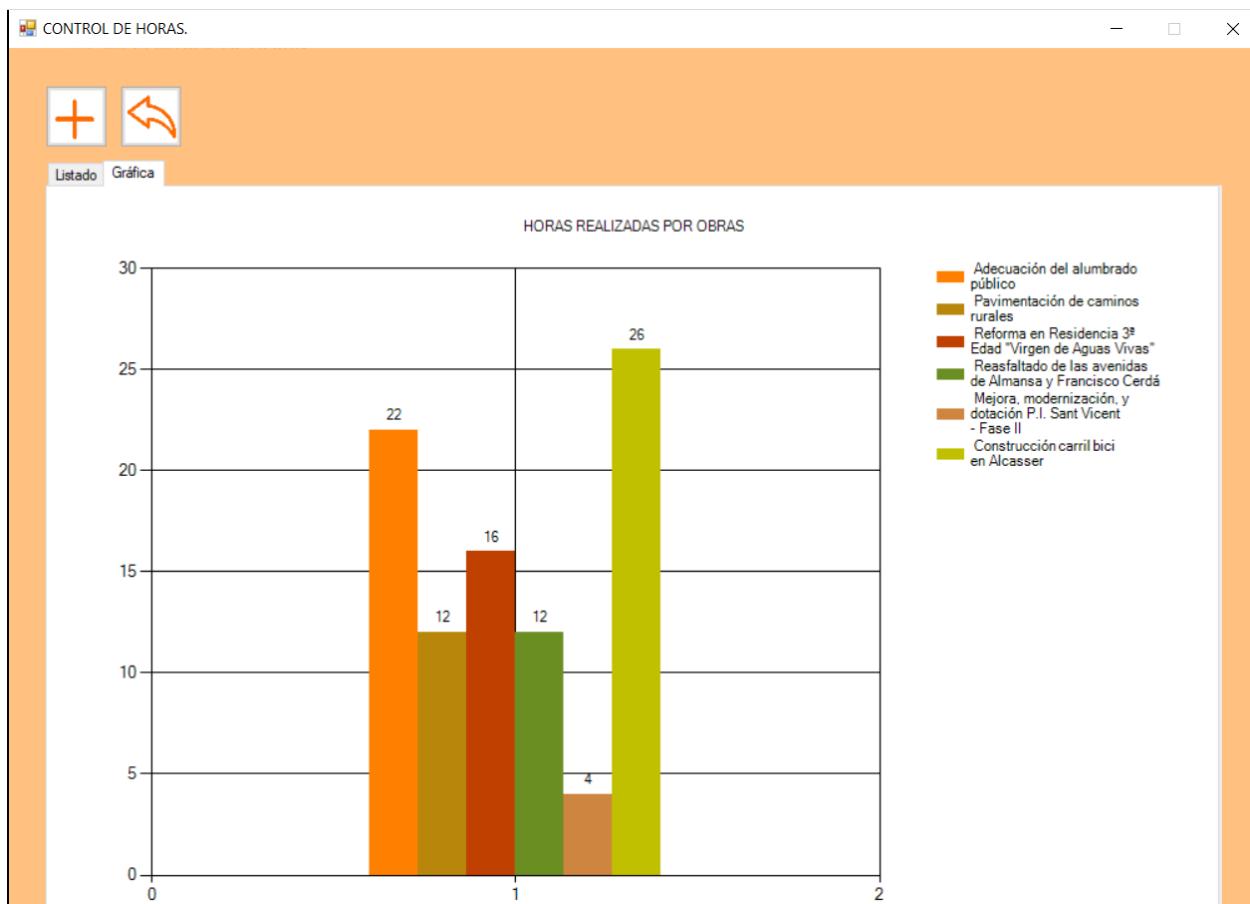
CONTROL DE HORAS.



+

Listado Gráfica

Código	Fecha	Obra	Expediente	Descripción	Entrada	Salida	Total
1	2021-05-21	1	348/2021bis	Adecuación del alumbrado público	09:00	14:00	05:00:00
2	2021-05-24	3	SIMPPAVCAMR...	Pavimentación de caminos rurales	09:30	15:30	06:00:00
3	2021-05-25	3	SIMPPAVCAMR...	Pavimentación de caminos rurales	07:30	13:30	06:00:00
4	2021-05-26	5	G-7619/2020	Reforma en Residencia 3º Edad "Vir...	07:30	15:30	08:00:00
5	2021-05-27	2	4289/2021	Reasfaltado de las avenidas de Alma...	08:45	14:00	05:15:00
6	2021-05-27	7	5066/2021	Mejora, modernización, y dotación P.I...	15:30	19:30	04:00:00
7	2021-05-28	1	348/2021bis	Adecuación del alumbrado público	09:30	14:30	05:00:00
8	2021-05-31	1	348/2021bis	Adecuación del alumbrado público	06:30	12:30	06:00:00
9	2021-06-01	5	G-7619/2020	Reforma en Residencia 3º Edad "Vir...	08:45	14:45	06:00:00
10	2021-06-02	2	4289/2021	Reasfaltado de las avenidas de Alma...	08:30	14:30	06:00:00
11	2021-06-02	5	G-7619/2020	Reforma en Residencia 3º Edad "Vir...	11:50	13:50	02:00:00
12	2021-07-15	7	363/2021	Construcción camí bici en Alcasser	06:00	14:00	08:00:00
13	2021-06-03	7	363/2021	Construcción camí bici en Alcasser	05:00	14:00	09:00:00
14	2021-06-03	7	363/2021	Construcción camí bici en Alcasser	05:00	14:00	09:00:00
15	2021-06-04	2	4289/2021	Reasfaltado de las avenidas de Alma...	10:30	11:30	01:00:00
16	2021-06-03	1	348/2021bis	Adecuación del alumbrado público	08:00	14:00	06:00:00



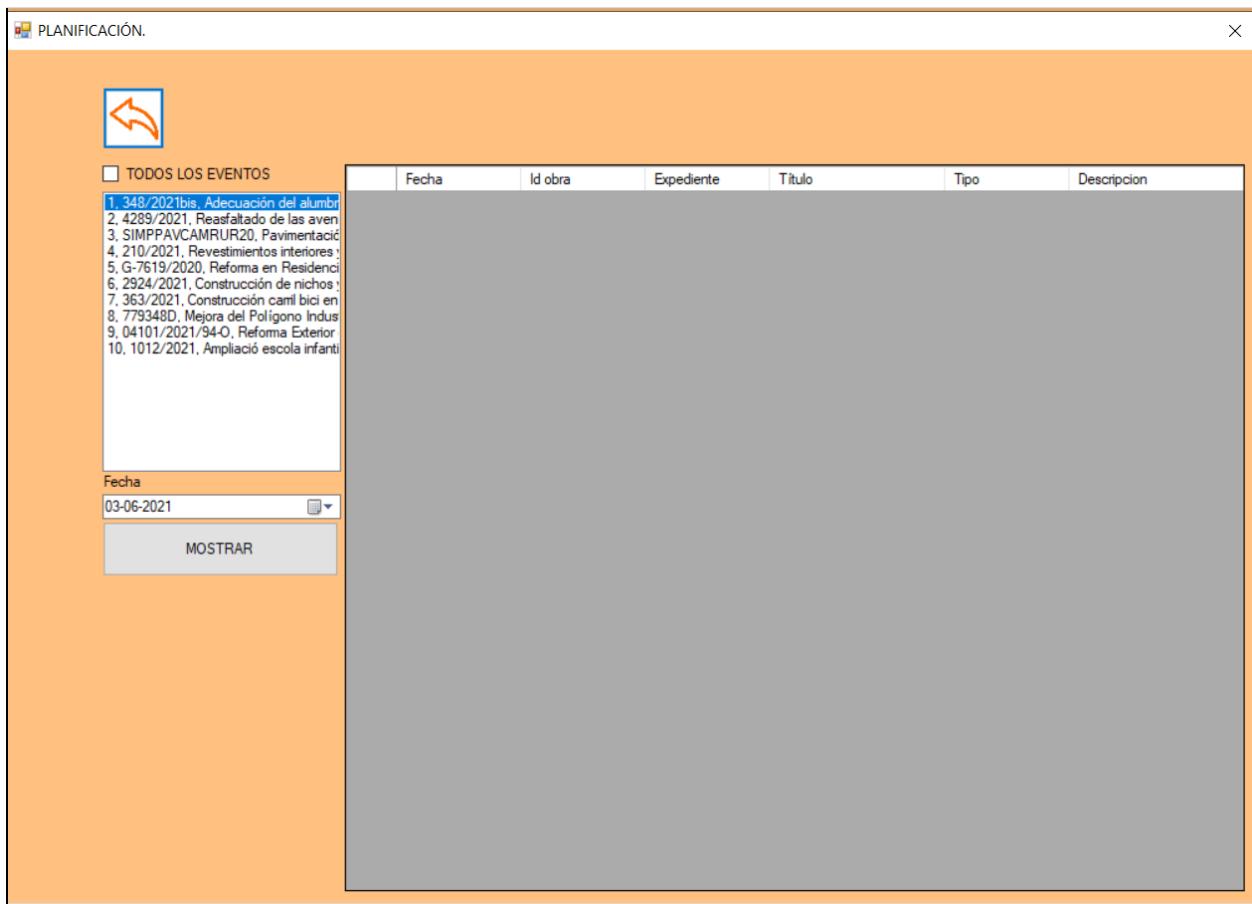
En este caso, al pulsar “+” el formulario de “Nuevo registro HORAS” nos solicitará la fecha, la obra y las horas de entrada y salida. Al pulsar “GUARDAR” se calculará automáticamente el total de horas realizadas que no será modificable.

Nuevo registro HORAS.

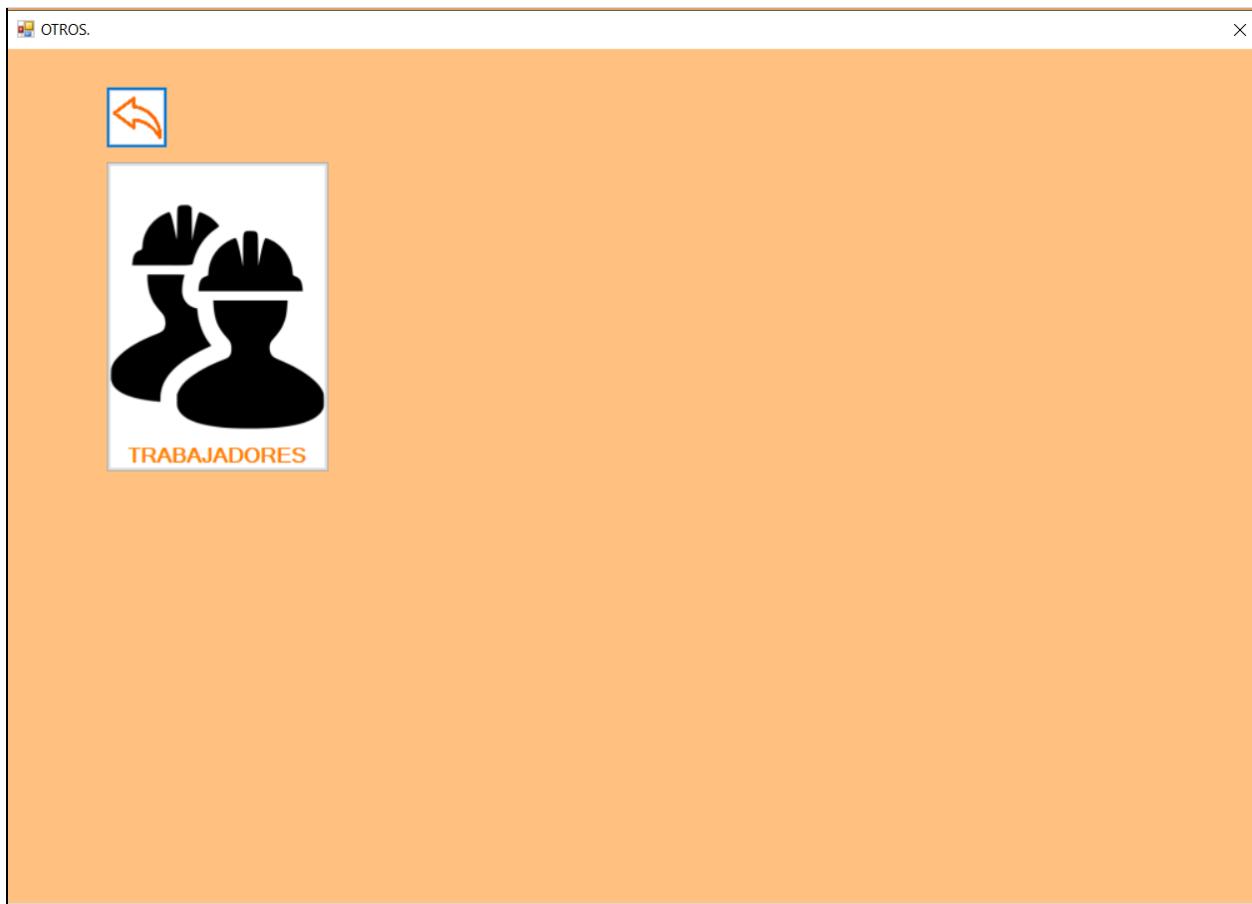
Código	17
Fecha	2021-06-03
Obra	1. 348/2021bis. Adecuación del alumbrado público 2. 4289/2021, Reasfaltado de las avenidas de Almansa y Frar 3. SIMPPAVCAMRUR20. Pavimentación de caminos rurales 5. G-7619/2020, Reforma en Residencia 3º Edad "Virgen de J 6. 2924/2021, Construcción de nichos y urbanización interior C 7. 363/2021, Construcción camil bici en Alcasser 8. 779348D, Mejora del Polígono Industrial Alqueria de Raga
H. Entrada	<input type="text"/>
H. Salida	<input type="text"/>
Total	<input type="text"/>

GUARDAR

Si volvemos al menú principal, podemos acceder desde el tercer botón “PLANIFICACIÓN” a la siguiente ventana. En ella, podemos encontrar el botón salir y, debajo de este, una serie de campos. Primero deberemos seleccionar si queremos ver todos los eventos o, por el contrario, solamente los eventos referentes a un expediente en concreto. Después, seleccionaremos la fecha y, por último, el botón “MOSTRAR”. Si existen eventos para la/s obra/s y fecha seleccionada nos aparecerán en el listado de la derecha.



La última opción del menú “OTROS” nos dirige al submenú de “TRABAJADORES”:



Al acceder, nos aparece una nueva ventana, con los botones de “+” y de salir. Y, también, un listado con los trabajadores existentes.

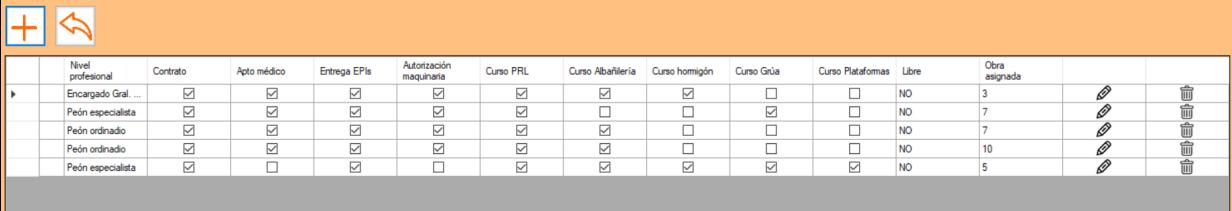
TRABAJADORES.														
Código empleado	Nombre	Apellidos	DNI	Teléfono	Nivel profesional	Contrato	Apto médico	Entrega EPs	Autorización maquinaria	Curso PRL	Curso Albañilería	Curso hormigón	Curso Guía	Curse
1	Manuel	García López	00000000A	600000000	Encargado Gral. ...	<input checked="" type="checkbox"/>	<input type="checkbox"/>							
2	José	Moreno Navarro	11111111B	611111111	Peón especialista	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						
3	Carmen	Gil Ramos	22222222C	622222222	Peón ordinario	<input checked="" type="checkbox"/>	<input type="checkbox"/>							
4	Sonia	Ureña Martínez	33333333D	633333333	Peón ordinario	<input checked="" type="checkbox"/>	<input type="checkbox"/>							
5	Pepe	dasd	0000000	6666	Peón especialista	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Se pueden crear nuevos trabajadores desde el botón “+”. El código del empleado no puede modificarse, el resto de los campos son datos personales, si el trabajador está o no libre y, en su caso, la obra en la que está trabajando. Asimismo se debe seleccionar el nivel profesional, la documentación recogida de dicho empleado y los cursos de los que dispone. Se seleccionará “GUARDAR” antes de salir para registrar al nuevo trabajador en la base de datos.

 Nuevo TRABAJADOR X

Código	6	Apellidos	
Nombre		LIBRE	SI NO
DNI		Obra	1. 348/2021bis, Adecuación del alumbrado 2. 4289/2021, Reasfaltado de las avenidas 3. SIMPPAVCAMRUR20, Pavimentación d 4. 210/2021, Revestimientos interiores y ur 5. G-7619/2020, Reforma en Residencia 3 6. 2924/2021, Construcción de nichos y ur 7. 363/2021, Construcción carretera bici en Alc
Teléfono			
Nivel prof.	Encargado Gral. Obra Oficial 1 ^a Oficial 2 ^a Peón especialista Peón ordinario Aprendiz		
Documentación		Cursos realizados	
<input type="checkbox"/> Contrato de trabajo firmado <input type="checkbox"/> Entrega de EPI's <input type="checkbox"/> Autorización uso maquinaria <input type="checkbox"/> APTO revisión médica		<input type="checkbox"/> PRL <input type="checkbox"/> Grúa <input type="checkbox"/> Albañilería <input type="checkbox"/> Plataformas <input type="checkbox"/> Hormigón	
1 348/2021bis Adecuación del alu		GUARDAR	

Desde el listado y, al final de cada línea, se encontrarán los botones de editar (lápiz) y eliminar (papelera):



El listado muestra una tabla con los siguientes datos:

Nivel profesional	Contrato	Apto médico	Entrega EPI's	Autorización maquinaria	Curso PRL	Curso Albañilería	Curso hormigón	Curso Grúa	Curso Plataformas	Libre	Obra asignada	Editar	Eliminar	
Encargado Gral. ...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NO	3								
Peón especialista	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NO	7						
Peón ordinario	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NO	7							
Peón ordinario	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NO	10							
Peón especialista	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NO	5		

Si pulsamos sobre el de editar, se podrán hacer algunos cambios en el trabajador como modificar el número de teléfono, dejarlo o no libre, cambiar su nivel profesional y añadir/eliminar documentación y cursos.

Nuevo TRABAJADOR

Código	5	Apellidos	dasd
Nombre	Pepe	LIBRE	SI NO
DNI	0000000	Obra	1, 348/2021bis, Adecuación del alumbrado 2, 4289/2021, Reasfaltado de las avenidas 3, SIMPPAVCAMRUR20, Pavimentación d 4, 210/2021, Revestimientos interiores y ur 5, G-7619/2020, Reforma en Residencia 3 6, 2924/2021, Construcción de nichos y ur 7, 363/2021, Construcción camí bici en Alc
Teléfono	6666		
Nivel prof.	Encargado Gral. Obra Oficial 1 ^a Oficial 2 ^a Peón especialista Peón ordinario Aprendiz		
Documentación		Cursos realizados	
<input checked="" type="checkbox"/> Contrato de trabajo firmado <input type="checkbox"/> Entrega de EPI's <input checked="" type="checkbox"/> Autorización uso maquinaria <input type="checkbox"/> APTO revisión médica		<input checked="" type="checkbox"/> PRL <input checked="" type="checkbox"/> Grúa <input checked="" type="checkbox"/> Albañilería <input checked="" type="checkbox"/> Plataformas <input checked="" type="checkbox"/> Hormigón	
5 G-7619/2020 Reforma en Reside		GUARDAR	

Si se pulsa el botón eliminar, se suprimirá al trabajador del listado y de la base de datos, previamente, se nos mostrará un mensaje para confirmar si se desea o no quitar al trabajador definitivamente.

