

# Documentacion reportes

## Tabla de contenido

---

BasedeDatos .....	3
CarpetaReportes1 .....	4
Formulario Crud .....	5
DescargarHelpNDoc .....	5
Documentacion .....	6
FormHTML .....	18
PDF .....	19
ReportedeFacturas .....	19

## BasedeDatos

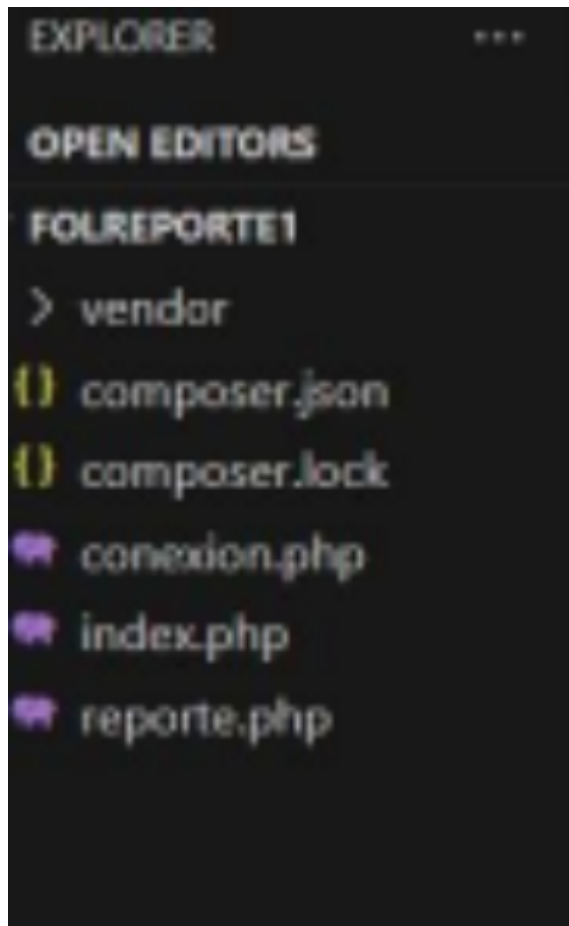
.apMasiel.Factura - dbo.Factura X SQLQuery3.sql - L...MASIEL\miaun (58))* SQLQuery2.sql - L...MASIEL							
	ID	DESCRIPC...	CATEGORIA	CANTIDAD	PRECIO_UN...	ITEBIS	DESCUENTO
▶	1	Laptop Len...	Electrónica	2	35000.00	0.18	0.05
	2	Camisa Polo	Ropa	5	800.00	0.18	0.00
	3	Celular Sam...	Tecnología	1	21000.00	0.18	0.10
	4	Silla ergonó...	Muebles	3	7500.00	0.18	0.08
	5	Auriculares ...	Accesorios	4	2500.00	0.18	0.02
	6	hola	adad	3	340.00	12.00	3.00
	7	kola real	supreme	10	1000.00	18.00	15.00
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE Factura (
    ID INT PRIMARY KEY IDENTITY(1,1),
    DESCRIPCION VARCHAR(100),
    CATEGORIA VARCHAR(50),
    CANTIDAD INT,
    PRECIO_UNITARIO DECIMAL(10,2),
    ITEBIS DECIMAL(10,2),
    DESCUENTO DECIMAL(10,2)
);
```

Creado con el Personal Edition de HelpNDoc: [Haz el cambio a CHM con la sencilla herramienta de conversión WinHelp HLP a CHM de HelpNDoc](#)

## CarpetaReportes1

---



---

Creado con el Personal Edition de HelpNDoc: [Transforma tu documento de Word en un libro electrónico de calidad profesional con HelpNDoc](#)

---

## Formulario Crud

The screenshot shows a web application window titled 'Form1'. It contains a form with the following fields and buttons:

- Form Fields:**
  - Descripcion (text input)
  - Cantidad (text input)
  - ID (text input)
  - Categoria (text input)
  - Precio (text input)
  - ITBIS (text input)
  - Descuento (text input)
- Buttons:**
  - Crear
  - Editar
  - Eliminar
  - Buscar
  - Reporte
- Data Table:**

	ID	DESCRIPCION	CATEGORIA	CANTIDAD	PRECIO_UNITARIO	ITEBIS	DESCU
▶	1	Laptop Lenovo	Electrónica	2	35000.00	0.18	0.05
	2	Camisa Polo	Ropa	5	800.00	0.18	0.00
	3	Celular Samsung ...	Tecnología	1	21000.00	0.18	0.10
	4	Silla ergonómica	Muebles	3	7500.00	0.18	0.08
	5	Auriculares JBL	Accesorios	4	2500.00	0.18	0.02
	6	hola	edad	3	340.00	12.00	3.00
	7	kola real	supreme	10	1000.00	18.00	15.00
*							

Creado con el Personal Edition de HelpNDoc: [No te quedes en el pasado: convierte tus archivos de ayuda WinHelp HLP a CHM con HelpNDoc](#)

## Descargar HelpNDoc

[Funciones](#)
[Descarga](#)
[Precios](#)

BLOG AYUDA CONTACTO

### Descargar gratuitamente HelpNDoc Edición Personal

#### Versión gratuita de HelpNDoc

**Descarga gratuita**  
 Completamente gratuito para uso personal »  
 v9.9.0.637 49.6MB

HelpNDoc es gratuito para uso personal y de evaluación

Descarga la versión gratuita de HelpNDoc para **crear con facilidad increíbles libros, archivos de ayuda, manuales de usuario y documentaciones** generando archivos de ayuda CHM, páginas web HTML dinámicas, documentos PDF y Word, páginas web móviles para iPhone y Android, eBooks ePub y Kindle, documentos Markdown y archivos Qt Help sin esfuerzo... ¡Y mucho más!

HelpNDoc está disponible de manera **completamente gratuita para uso personal y de evaluación**: podrás usar la versión completa de HelpNDoc durante todo el tiempo que quieras. La mayoría de las funciones están disponibles y habilitadas, no hay cuotas ocultas ni proceso de inscripción y no contiene ninguna clase de virus, spyware o malware. La única restricción es que ni HelpNDoc Edición Personal ni los archivos generados con ella pueden usarse con ánimo de lucro: esta licencia sólo permite el trabajo no lucrativo para uso personal o de evaluación.

Creado con el Personal Edition de HelpNDoc: [Crear archivos de ayuda para Qt Help Framework](#)

## Documentacion

# Documentacion Creacion de Reportes en c# y PHP

## ☐ Información General

### Base de Datos

**Estudiante:** Hasley Mariel Cuevas Forch

**Fecha:** 05/06/2025

**Profesor:** Victor Vladimir Recio

## Datos usados

En este proyecto se utilizaron **dos bases de datos diferentes**, según el lenguaje:

- **PHP:** se conecta a una base de datos en **MySQL/MariaDB** creada con **HeidiSQL**.
- **C#:** se conecta a una base de datos llamada **Factura** en **SQL Server**.

Ambas bases contienen una tabla llamada `Factura` que almacena la información detallada de cada factura

---

## Introducción

Esta documentación describe la implementación y el paso a paso para hacer informes en **C#** y **PHP** para una aplicación de facturación. Los informes son una parte fundamental de las aplicaciones de software, ya que permiten visualizar la información de manera organizada, clara y útil para el usuario final. En este proyecto, se utilizan diferentes tecnologías para generar reportes en ambos lenguajes: **C# con ReportViewer** y **PHP con FPDF**.

## Estructura de la tabla `Factura`

La tabla `Factura` contiene los siguientes campos:

1. **ID** (`INT`, Clave Primaria): Identificador único de la factura.
2. **DESCRIPCION** (`VARCHAR(255)`): Descripción del producto o servicio facturado.
3. **CATEGORIA** (`VARCHAR(100)`): Categoría del producto o servicio.
4. **CANTIDAD** (`INT`): Cantidad del producto o servicio facturado.
5. **PRECIO\_UNITARIO** (`DECIMAL(10,2)`): Precio por unidad del producto o servicio.
6. **ITEBIS** (`DECIMAL(10,2)`): Impuesto aplicado a la factura (ITBIS).
7. **DESCUENTO** (`DECIMAL(10,2)`): Descuento aplicado al total.

## Creamos la base de datos

```
CREATE DATABASE BD_FacturacionPruebas;
GO
```

Usando la base de datos ya creada, creamos sus campo

```
USE BD_FacturacionPruebas;

CREATE TABLE Factura (
    ID INT IDENTITY PRIMARY KEY,
    DESCRIPCION VARCHAR(255),
    CATEGORIA VARCHAR(100),
    CANTIDAD INT,
    PRECIO_UNITARIO DECIMAL(10,2),
    ITEBIS DECIMAL(10,2),
    DESCUENTO DECIMAL(10,2),
    TOTAL_GENERAL DECIMAL(10,2)
);
GO
```

Usando la base datos, ahora insertamos algunos registros

```
USE BD_FacturacionPruebas;

INSERT INTO Factura (DESCRIPCION, CATEGORIA, CANTIDAD, PRECIO_UNITARIO, ITEBIS, DESCUENTO,
VALUES
    ('Mouse inalámbrico', 'Electrónica', 3, 350.00, 189.00, 50.00, 1189.00),
    ('Silla ergonómica', 'Mobiliario', 2, 2200.00, 792.00, 100.00, 5292.00),
    ('Memoria USB 32GB', 'Accesorios', 5, 250.00, 225.00, 0.00, 1500.00),
    ('Monitor LED 24"', 'Electrónica', 1, 7800.00, 1404.00, 300.00, 8904.00);
```

## Reporte en c# con reportviewer

En C#, los informes se pueden generar de distintas formas, sin embargo, en este caso, lo crearemos utilizando ReportViewer

- **Instalación del paquete NuGet**

Se requiere la instalación del paquete NuGet

[Microsoft.ReportingServices.ReportViewerControl.WinForms](#).

Esto se puede hacer desde el Administrador de paquetes NuGet en Visual Studio.

## Archivos C#

Se crearon los siguientes archivos:

**Form1.cs:** Formulario para mostrar el informe.

**Crud.cs:** Formulario para el CRUD de facturas.

**reporte.rdlc:** Archivo de diseño del reporte.

## Codigo C# formreporte(Form1)

Este formulario carga y muestra el reporte de facturas utilizando el control ReportViewer. Ten en cuenta que algunos factores cambian según cada usuario.

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

namespace reportefol1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            MostrarDatos();
            string connectionString = "Server=localhost;Database=Factura;Integrated Security=True;";
            string query = "SELECT * FROM Factura";
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
            }
        }

        private void Btn_Reporte_Click(object sender, EventArgs e)
        {
            reporte reporteForm = new reporte();
            reporteForm.ShowDialog();
        }

        private void Btn_crear_Click(object sender, EventArgs e)
        {
            MostrarDatos();
            string connectionString = "Server=localhost;Database=Factura;Integrated Security=True;";
            string query = "INSERT INTO Factura (DESCRIPCION, CATEGORIA, CANTIDAD, PRECIO_UN
                "VALUES (@descripcion, @categoria, @cantidad, @precio, @itebis, @descuento)";
            // Convertir valores
            int cantidad;
            decimal precioUnitario, itebis, descuento;
            if (!int.TryParse(textcantidad.Text, out cantidad) ||
                !decimal.TryParse(textprecio.Text, out precioUnitario) ||
                !decimal.TryParse(textitbis.Text, out itebis) ||
```



```

        !decimal.TryParse(txtdescuento.Text, out descuento))
    {
        MessageBox.Show("Verifica que los campos numéricos sean válidos.");
        return;
    }

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        try
        {
            SqlCommand command = new SqlCommand(query, connection);
            command.Parameters.AddWithValue("@descripcion", textdescripcion.Text);
            command.Parameters.AddWithValue("@categoria", textcategoria.Text);
            command.Parameters.AddWithValue("@cantidad", cantidad);
            command.Parameters.AddWithValue("@precio", precioUnitario);
            command.Parameters.AddWithValue("@itebis", itebis);
            command.Parameters.AddWithValue("@descuento", descuento);
            connection.Open();
            int filas = command.ExecuteNonQuery();
            if (filas > 0)
            {
                MessageBox.Show("Factura registrada correctamente.");
                LimpiarCampos1();
            }
            else
            {
                MessageBox.Show("No se pudo insertar la factura.");
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error: " + ex.Message);
        }
    }
}

private void LimpiarCampos1()
{
    textdescripcion.Clear();
    textcategoria.Clear();
    textcantidad.Clear();
    textprecio.Clear();
    textitbis.Clear();
    txtdescuento.Clear();
}

private void Btn_Eliminar_Click(object sender, EventArgs e)
{
    MostrarDatos();
    string connectionString = "Server=localhost;Database=Factura;Integrated Security=True;";
    string query = "DELETE FROM Factura WHERE ID = @id";
    int id;
    if (!int.TryParse(textID.Text, out id))

```

```

{
    MessageBox.Show("Por favor, introduce un ID válido.");
    return;
}

using (SqlConnection connection = new SqlConnection(connectionString))
{
    try
    {
        SqlCommand command = new SqlCommand(query, connection);
        command.Parameters.AddWithValue("@id", id);
        connection.Open();
        int filas = command.ExecuteNonQuery();
        if (filas > 0)
        {
            MessageBox.Show("Factura eliminada correctamente.");
            LimpiarCampos();
        }
        else
        {
            MessageBox.Show("No se encontró una factura con ese ID.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex.Message);
    }
}

private void LimpiarCampos()
{
    textdescripcion.Clear();
    textcategoria.Clear();
    textcantidad.Clear();
    textprecio.Clear();
    textitbis.Clear();
    txtdescuento.Clear();
    textID.Clear();
}

private void Btn_Editar_Click(object sender, EventArgs e)
{
    MostrarDatos();
    string connectionString = "Server=localhost;Database=Factura;Integrated Security=True;";
    string query = @"UPDATE Factura
        SET DESCRIPCION = @descripcion,
            CATEGORIA = @categoria,
            CANTIDAD = @cantidad,
            PRECIO_UNITARIO = @precio,
            ITEBIS = @itebis,
            DESCUENTO = @descuento,
            WHERE ID = @id";

```

```

int id;
if (!int.TryParse(textID.Text, out id))
{
    MessageBox.Show("Introduce un ID válido.");
    return;
}

try
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        command.Parameters.AddWithValue("@id", id);
        command.Parameters.AddWithValue("@descripcion", textdescripcion.Text);
        command.Parameters.AddWithValue("@categoria", textcategoria.Text);
        command.Parameters.AddWithValue("@cantidad", Convert.ToInt32(textcantidad.Text));
        command.Parameters.AddWithValue("@precio", Convert.ToDecimal(textprecio.Text));
        command.Parameters.AddWithValue("@itebis", Convert.ToDecimal(textitbis.Text));
        command.Parameters.AddWithValue("@descuento", Convert.ToDecimal(txtdescuento.Text));

        decimal precio = Convert.ToDecimal(textprecio.Text);
        int cantidad = Convert.ToInt32(textcantidad.Text);
        decimal itbis = Convert.ToDecimal(textitbis.Text);
        decimal descuento = Convert.ToDecimal(txtdescuento.Text);

        connection.Open();
        int rows = command.ExecuteNonQuery();
        if (rows > 0)
        {
            MessageBox.Show("Factura actualizada correctamente.");
            LimpiarCampos();
        }
        else
        {
            MessageBox.Show("No se encontró una factura con ese ID.");
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show("Error al actualizar: " + ex.Message);
}
}

private void Btn_Buscar_Click(object sender, EventArgs e)
{
    MostrarDatos();
    string connectionString = "Server=localhost;Database=Factura;Integrated Security=True;";
    string query = "SELECT * FROM Factura WHERE ID = @id";
    int id;
    if (!int.TryParse(textID.Text, out id))
    {
        MessageBox.Show("Introduce un ID válido.");
    }
}

```

```

        return;
    }

    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@id", id);
            connection.Open();
            using (SqlDataReader reader = command.ExecuteReader())
            {
                if (reader.Read())
                {
                    textdescripcion.Text = reader["DESCRIPCION"].ToString();
                    textcategoria.Text = reader["CATEGORIA"].ToString();
                    textcantidad.Text = reader["CANTIDAD"].ToString();
                    textprecio.Text = reader["PRECIO_UNITARIO"].ToString();
                    textitbis.Text = reader["ITEBIS"].ToString();
                    txtdescuento.Text = reader["DESCUENTO"].ToString();
                }
                else
                {
                    MessageBox.Show("No se encontró una factura con ese ID.");
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error al buscar: " + ex.Message);
    }
}

private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    // Aquí puedes poner lógica si deseas usar los datos de una fila al hacer clic
}

private void MostrarDatos()
{
    string connectionString = "Server=localhost;Database=Factura;Integrated Security=True;";
    string query = "SELECT ID, DESCRIPCION, CATEGORIA, CANTIDAD, PRECIO_UNITARIO, ITEBIS, DESCUENTO";

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        try
        {
            SqlDataAdapter adaptador = new SqlDataAdapter(query, connection);
            DataTable tabla = new DataTable();
            adaptador.Fill(tabla);
            dataGridView1.DataSource = tabla;
        }
    }
}

```

```

        catch (Exception ex)
        {
            MessageBox.Show("Error al cargar los datos: " + ex.Message);
        }
    }
}
}
}

```

## Informe RDLC

Este archivo es el diseño del informe en C#. Aquí se define la estructura visual del informe, incluyendo los campos que se mostrarán (como ID, Descripción, Categoría, etc.) y su formato. También se configura la conexión a la base de datos y la tabla de la cual se obtendrán los datos.

Luego en " ReporteForm.cs " (como ya vimos previamente en la parte de arriba) en el botón de cargar informe, se especifica la ruta del archivo " ReporteFactura.rdlc " para que se muestre en el " ReportViewer ".

- Ahora vamos a ver como se hace un informe de PHP.

## Informe php con dompdf

Para generar informes en PHP, en este caso, se utilizó la librería Dompdf.

1. Abra la línea de comandos y navegue hasta la carpeta del proyecto (donde se encuentra el archivo `composer.json`).
2. Ejecutar el siguiente comando:

### Instalar con git

Desde la línea de comandos, cambie al directorio donde residirá dompdf y ejecute los siguientes comandos:

```

git clone https://github.com/dompdf/dompdf.git
cd dompdf/lib

git clone https://github.com/PhenX/php-font-lib.git php-font-lib
cd php-font-lib
git checkout 0.5.1
cd ..

git clone https://github.com/PhenX/php-svg-lib.git php-svg-lib
cd php-svg-lib
git checkout v0.3.2
cd ..

git clone https://github.com/sabberworm/PHP-CSS-Parser.git php-css-parser
cd php-css-parser
git checkout 8.1.0

```

Esto instalará Dompdf y sus dependencias en la carpeta `vendor` del proyecto.

## Instalación de composer

Primero, se debe instalar Composer, que es un gestor de dependencias para PHP.

 Inicio | [Empezando](#) | [Descargar](#) | [Documentación](#) | [Explorar paquetes](#)

### Descargar Composer Última versión: v2.8.9

#### Instalador de Windows

El instalador, que requiere que ya tenga PHP instalado, descargará Composer para usted y configurará su variable de entorno PATH para que pueda llamarlo simplemente `composer` desde cualquier directorio.

Descargue y ejecute [Composer-Setup.exe](#) : instalará la última versión de Composer cada vez que se ejecute.

#### Instalación desde la línea de comandos

Para instalar rápidamente Composer en el directorio actual, ejecute el siguiente script en su terminal. Para automatizar la instalación, consulte [la guía sobre la instalación programática de Composer](#) .

```
php -r "copiar('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "si (archivo_hash('sha384', 'composer-setup.php') === 'dac665fdc30fdd8ec78b38b9800061b4150413ff2e3b6f885
php composer-setup.php
php -r "desvincular('composer-setup.php');"
```

Este script de instalación simplemente comprobará algunas `php.ini` configuraciones, le avisará si están configuradas incorrectamente y luego descargará la última versión `composer.phar` en el directorio actual. Las 4 líneas anteriores, en orden:

- 1.
1. Descargue y ejecute el instalador "Composer-Setup.exe" desde la página oficial. (Haga clic aquí para ir a la página oficial para instalar Composer)
- 2.
2. Siga las instrucciones del instalador, asegurándose de seleccionar la ruta correcta al archivo `php.exe` de XAMPP.
- 3.
3. Configure el proxy si es necesario (en este caso no) y finalice la instalación

### **Archivos PHP:**

La estructura de este proyecto PHP en particular incluye los siguientes archivos y

carpetas. Estos archivos permiten registrar facturas, administrarlas (crear, editar, eliminar) y generar reportes en formato PDF usando la librería FPDF

1. Conexion.php                      Conexión a la base de datos MySQL (HeidiSQL).
2. index.php                        Formulario HTML para registrar una factura.
3. reporte.php                      Muestra los registros en una tabla HTML.
4. fpdf.php                        Archivo principal de la librería FPDF.

## Código PHP. Formulario.html:

Este archivo HTML contiene el formulario con un botón que al ser presionado, llama al script `generar_reporte.php` para crear el informe.

```

index.php X  reporte.php  conexion.php
index.php
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Reporte de Facturas</title>
6      <style>
7          body {
8              font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
9              background-color: rgb(107, 46, 77);
10             margin: 0;
11             padding: 0;
12             display: flex;
13             justify-content: center;
14             align-items: center;
15             height: 100vh;
16         }
17
18         .contenedor {
19             background-color: rgb(246, 214, 229);
20             padding: 40px;
21             border-radius: 12px;
22             box-shadow: 0 4px 12px rgba(0,0,0,0.1);
23             text-align: center;
24             width: 400px;
25         }
26
27         h2 {
28             margin-bottom: 30px;
29             color: #333333;
30         }
31
32         button {
33             background-color: rgb(122, 42, 55);
34             color: white;
35             border: none;
36             padding: 12px 20px;
37             border-radius: 6px;

```

```

index.php X  reporte.php  conexion.php
index.php
3  <head>
6  <style>
32  button {
33      background-color: rgb(122, 42, 55);
34      color: white;
35      border: none;
36      padding: 12px 20px;
37      border-radius: 6px;
38      font-size: 16px;
39      cursor: pointer;
40      transition: background-color 0.3s ease;
41  }
42
43  button:hover {
44      background-color: #0056b3;
45  }
46
47  .icono {
48      font-size: 40px;
49      margin-bottom: 15px;
50      color: #007bff;
51  }
52  </style>
53  </head>
54  <body>
55  <div class="contenedor">
56      <div class="icono">📄</div>
57      <h2>Reporte de Facturas</h2>
58      <form action="reporte.php" method="post">
59          <button type="submit">📄 Generar PDF</button>
60      </form>
61  </div>
62  </body>
63  </html>
64

```

## Código PHP. Que genera el reporte:

Este script PHP es el encargado de generar el informe PDF utilizando la librería Dompdf.



```

index.php  reporte.php X  conexion.php
reporte.php
1  <?php
2  require 'vendor/autoload.php';
3  require 'conexion.php';
4
5  use Dompdf\Dompdf;
6
7  $html = '
8  <h1 style="text-align:center;">Reporte de Facturas</h1>
9  <table border="1" cellspacing="0" cellpadding="8" width="100%">
10     <thead style="background-color:#f2f2f2;">
11         <tr>
12             <th>ID</th>
13             <th>Descripción</th>
14             <th>Categoría</th>
15             <th>Cantidad</th>
16             <th>Precio Unitario</th>
17             <th>ITBIS</th>
18             <th>Descuento</th>
19             <th>Total</th>
20         </tr>
21     </thead>
22     <tbody>';
23
24 $sql = "SELECT * FROM facturas";
25 $resultado = $conn->query($sql);
26
27 while ($row = $resultado->fetch_assoc()) {
28     $total = ($row['Cantidad'] * $row['PrecioUnitario']) + $row['Itebis'] - $row['Descuento'];
29
30     $html .= '
31         <tr>
32             <td>' . $row['ID'] . '</td>
33             <td>' . $row['Descripcion'] . '</td>
34             <td>' . $row['Categoría'] . '</td>
35             <td>' . $row['Cantidad'] . '</td>
36             <td>' . number_format($row['PrecioUnitario'], 2) . '</td>
37             <td>' . number_format($row['Itebis'], 2) . '</td>

```

```

index.php  reporte.php X  conexion.php
reporte.php
9  <table border="1" cellspacing="0" cellpadding="8" width="100%">
21     </thead>
22     <tbody>';
23
24 $sql = "SELECT * FROM facturas";
25 $resultado = $conn->query($sql);
26
27 while ($row = $resultado->fetch_assoc()) {
28     $total = ($row['Cantidad'] * $row['PrecioUnitario']) + $row['Itebis'] - $row['Descuento'];
29
30     $html .= '
31         <tr>
32             <td>' . $row['ID'] . '</td>
33             <td>' . $row['Descripcion'] . '</td>
34             <td>' . $row['Categoría'] . '</td>
35             <td>' . $row['Cantidad'] . '</td>
36             <td>' . number_format($row['PrecioUnitario'], 2) . '</td>
37             <td>' . number_format($row['Itebis'], 2) . '</td>
38             <td>' . number_format($row['Descuento'], 2) . '</td>
39             <td><strong>' . number_format($total, 2) . '</strong></td>
40         </tr>';
41     }
42
43 $html .= '
44     </tbody>
45 </table>';
46
47 $dompdf = new Dompdf();
48 $dompdf->loadHtml($html);
49 $dompdf->setPaper('A4', 'landscape');
50 $dompdf->render();
51 $dompdf->stream("reporte_facturas.pdf", ["Attachment" => false]);
52 exit;

```

Para que la **generación de informes en PHP** funcione correctamente, es fundamental tener configurado un entorno de servidor.

En este caso, usamos **XAMPP**, por lo tanto:

-

- Asegúrate de que **Apache** y **MySQL** estén **activos** desde el Panel de Control de XAMPP.
- 
- Si alguno de estos servicios no está en ejecución, el código PHP no podrá conectarse a la base de datos ni generar los informes.

## Conclusión

Este proyecto muestra cómo realizar **informes automáticos** tanto en **C#** como en **PHP**, explicando paso a paso el código, configuración y ejecución.

Conocer la forma de crear reportes en distintos lenguajes y plataformas es súper útil porque:

- Nos permite **obtener datos de una base de forma rápida y ordenada**.
- Es una herramienta clave para presentar información útil en empresas, escuelas y proyectos.
- Mejora nuestra capacidad de análisis y toma de decisiones.

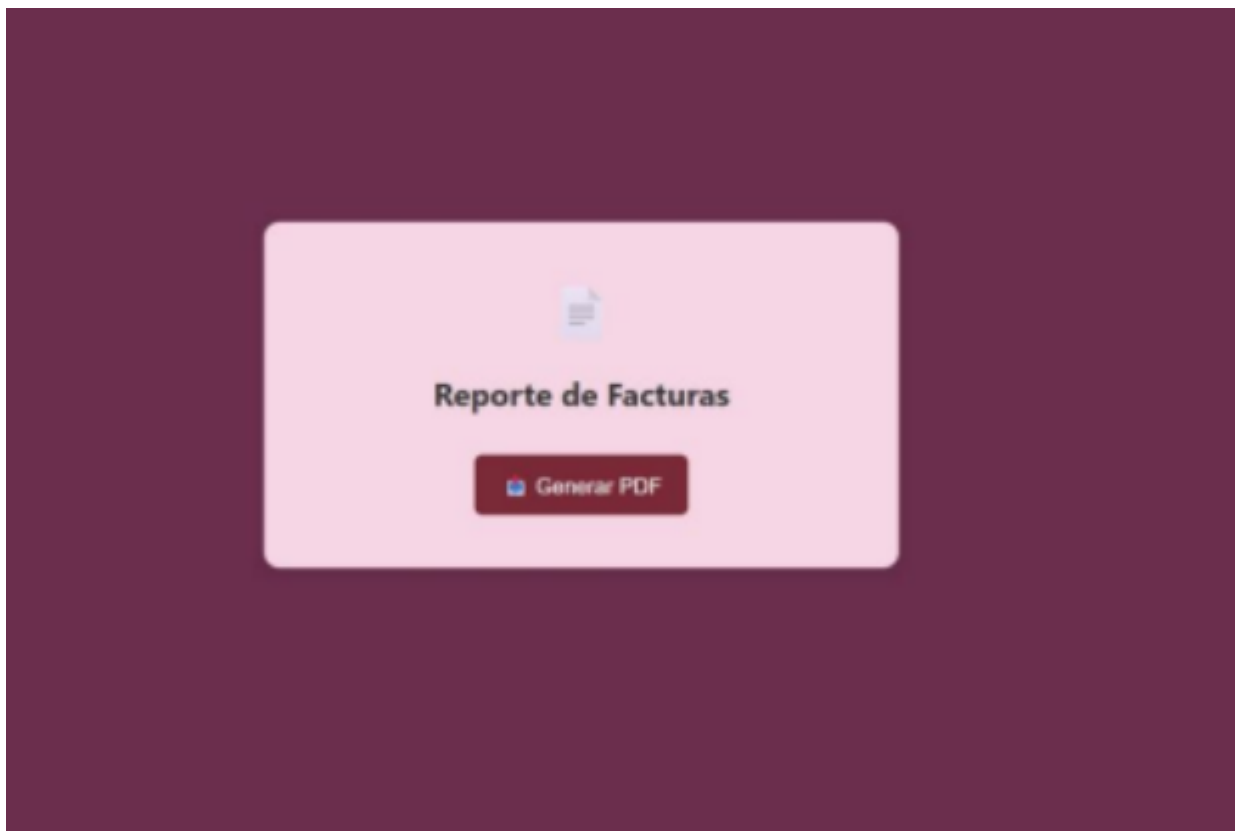
---

Creado con el Personal Edition de HelpNDoc: [Generar EBooks de EPub con facilidad](#)

---

## FormHTML

---



---

Creado con el Personal Edition de HelpNDoc: [Ventajas de las herramientas de creación de documentación de ayuda](#)

---

PDF

reporte.php

1 / 1100%

Reporte de Facturas

ID	Descripción	Categoría	Cantidad	Precio Unitario	ITBIS	Descuento	Total
1	Laptop Lenovo	Tecnología	2	\$1,200.00	\$216.00	\$108.00	\$2,516.00
2	Impresora HP	Oficina	1	\$450.00	\$81.00	\$50.00	\$481.00
3	Silla ergonómica	Muebles	3	\$150.00	\$81.00	\$0.00	\$531.00
4	Monitor Dell	Tecnología	2	\$300.00	\$108.00	\$20.00	\$688.00
5	Escritorio	Muebles	1	\$250.00	\$45.00	\$10.00	\$285.00
6	Teclado inalámbrico	Tecnología	4	\$75.00	\$54.00	\$10.00	\$344.00
7	Mouse inalámbrico	Tecnología	5	\$30.00	\$27.00	\$5.00	\$172.00
8	Archivador inalámbrico	Oficina	2	\$200.00	\$72.00	\$15.00	\$457.00
9	Routero WiFi	Tecnología	3	\$90.00	\$48.60	\$0.00	\$318.60
10	Líquido de escritorio	Muebles	4	\$40.00	\$28.80	\$8.00	\$180.80
11	Papel bond (copio)	Oficina	6	\$25.00	\$27.00	\$0.00	\$177.00
12	Proyector Epson	Tecnología	1	\$850.00	\$153.00	\$50.00	\$953.00
13	Gabinete de archivos	Muebles	1	\$300.00	\$54.00	\$25.00	\$329.00
14	Teléfono IP	Oficina	3	\$120.00	\$64.80	\$10.00	\$414.80
15	Cámara de seguridad	Tecnología	2	\$220.00	\$79.20	\$20.00	\$499.20

Creado con el Personal Edition de HelpNDoc: [Agilice su proceso de documentación con Project Analyzer de HelpNDoc](#)

ReportedeFacturas

reporte

1 de 1100%

Buscar

Siguiente

REPORTE DE FACTURAS

ID	CATEGORIA	CANTIDAD	DESCRIPCION	CATEGORIA	PRECIO UNITARIO	ITBIS	DESCUENTO	TOTAL GENERAL
66	Tecnologia	27	Laptop Dell Inspiron	Tecnologia	110640.00	27174.00	6453.00	231641.00

Creado con el Personal Edition de HelpNDoc: [Herramienta de creación de ayuda potente y fácil de usar para documentos Markdown](#)