HEINSOHN
BUSINESS TECHNOLOGY

Introducción a Visual Studio y C#

Microsoft Partner
Gold Software Development
Gold Web Development
Gold Software Asset Management

www.heinsohn.com.co

HEINSOHN
BUSINESS TECHNOLOGY

Agenda

- IDE Visual Studio
  - Administrando Proyectos y Soluciones
  - Herramientas
  - Creación de la primera aplicación
- Sintaxis de los lenguajes
  - Lógica de programación
  - Elección del lenguaje
  - Variables y tipos de datos
  - Estructuras lógicas

HEINSOHN
BUSINESS TECHNOLOGY

Visual Studio .NET

- VS.NET simplifica el desarrollo de aplicaciones basadas en .NET proporcionando un entorno de desarrollo simple y unificado

- Características
  - Un solo IDE (Integrated Development Environment)
  - Soporte para varios lenguajes .NET (VB.NET, C#,...)
  - Desarrollo de múltiples tipos de proyectos
  - Explorador Web integrado (basado en IE)
  - Interfase personalizable
  - Posee varias utilidades adicionales: Acceso a datos SQL Server, Depurador, Intellisense, Emuladores para móviles, etc.

HEINSOHN
BUSINESS TECHNOLOGY

Proyectos en Visual Studio .NET

- Los proyectos son contenedores que se ubican en una solución, para facilitar la administración, compilación y almacenado de los ítems
- Plantillas de Proyectos
  - Permiten la generación automática de módulos que serán administrados fácilmente, de manera tal que el desarrollador solo tenga que ocuparse de la funcionalidad específica.
- Archivos de definición de proyectos
  - Contienen metadata del proyecto
    - Ubicación física de los ítems
    - Forma de compilación
    - Archivos asociados

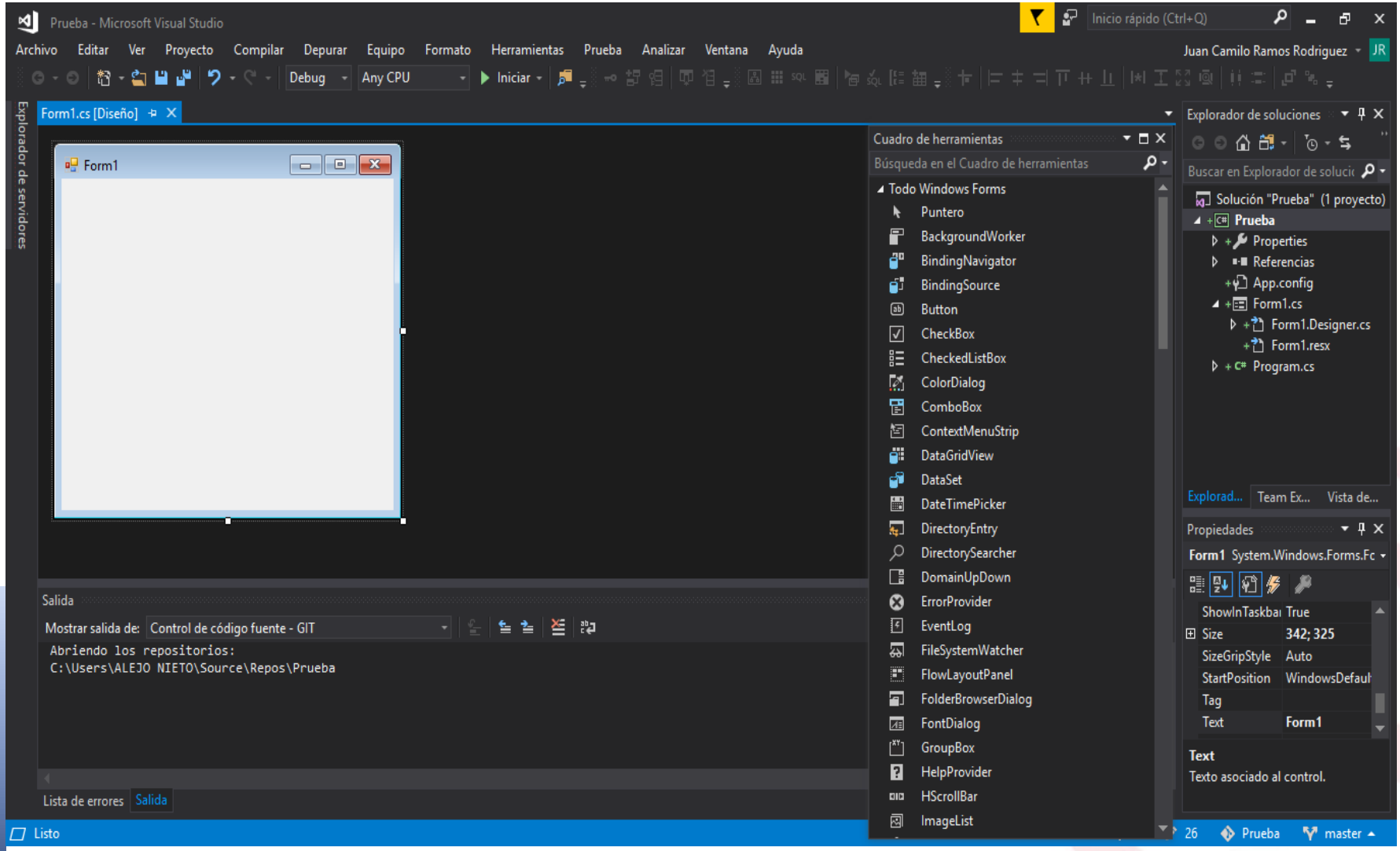
HEINSOHN
BUSINESS TECHNOLOGY

Soluciones en Visual Studio .NET

- Archivos o ítems relacionados se agrupan, a nivel lógico, en un proyecto
- Una solución puede contener varios proyectos.
  - Se pueden abrir, cerrar y guardar todos al mismo tiempo.
- La relación lógica entre solución y proyectos no refleja necesariamente una relación física.

HEINSOHN
BUSINESS TECHNOLOGY

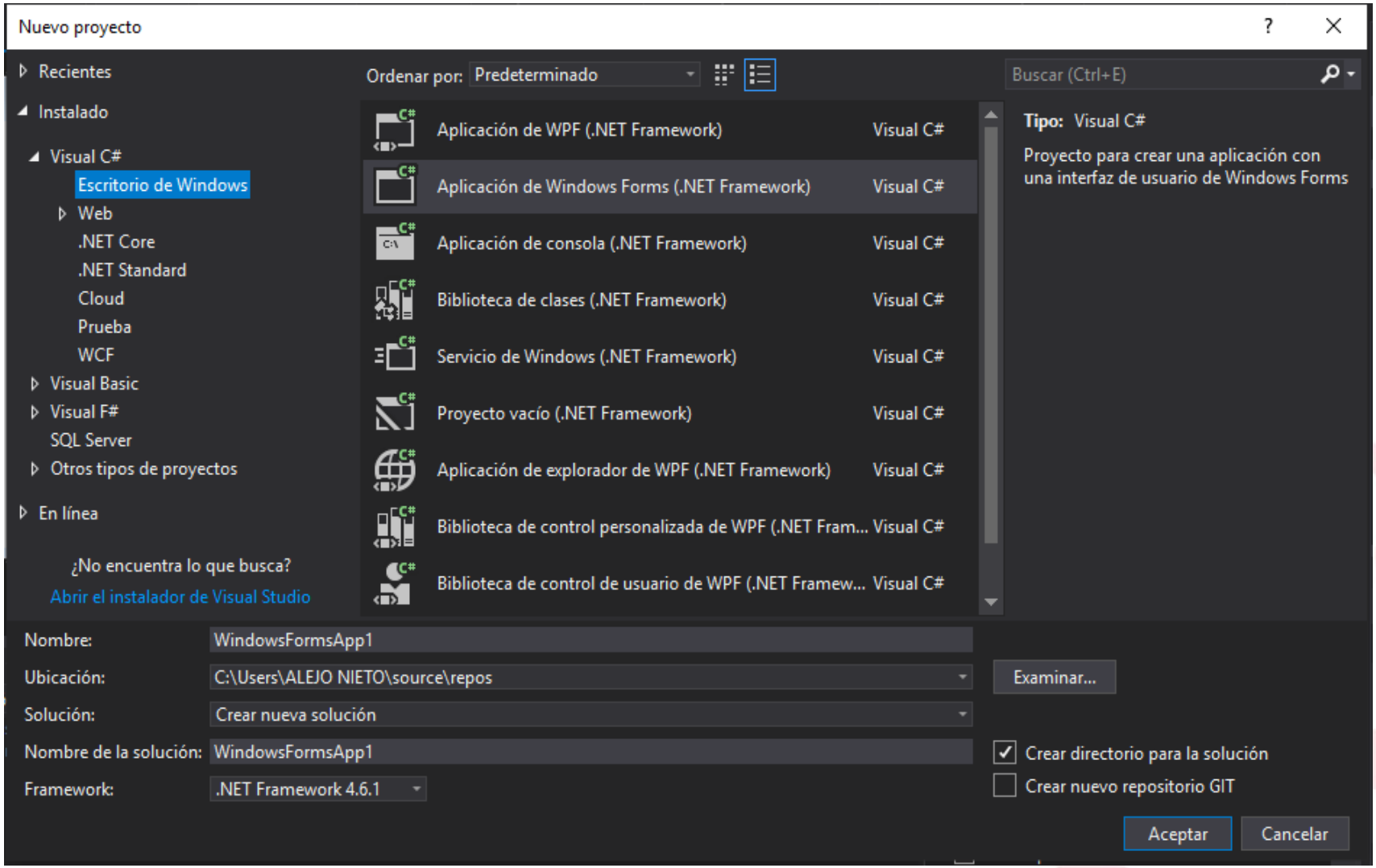
Visual Studio .NET IDE



Visual Studio IDE

HEINSOHN
BUSINESS TECHNOLOGY

Plantilla de Proyectos

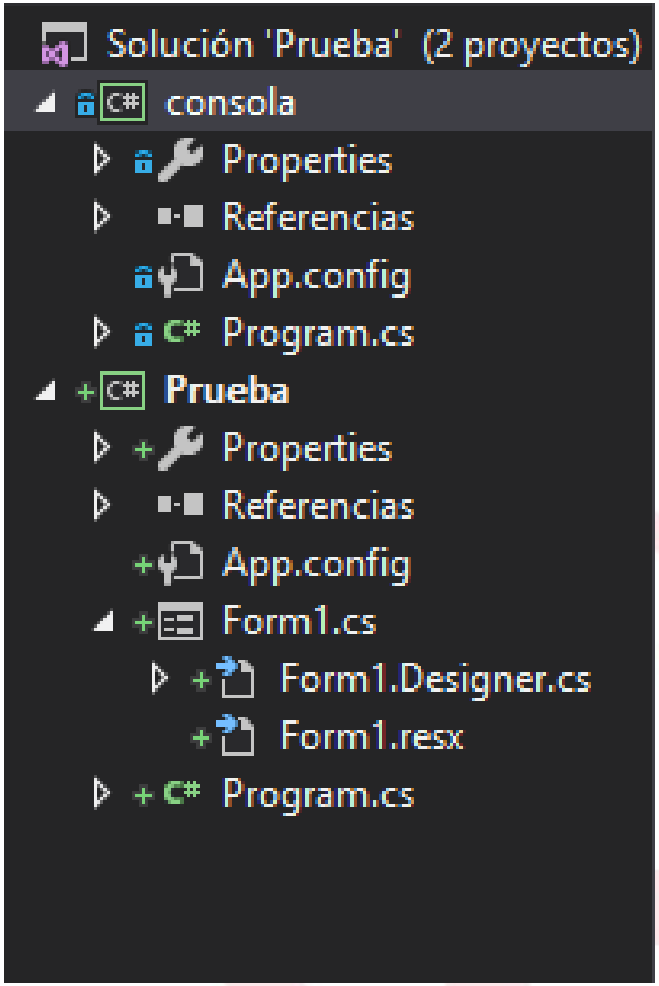


Visual Studio IDE

HEINSOHN
BUSINESS TECHNOLOGY

Solution Explorer

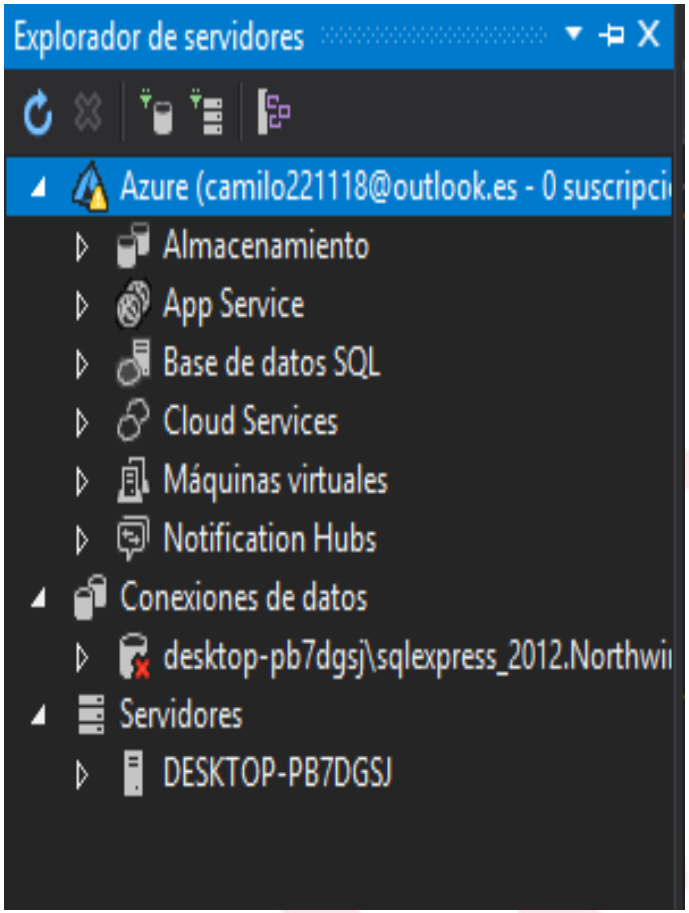
- Muestra los archivos de/los proyectos de la solución
- Permite eliminar y mover los archivos del proyecto
- Permite agregar nuevos elementos al proyecto
- Establecer referencias a asemblies y servicios web
- Crear carpetas
- Etc.



HEINSOHN
BUSINESS TECHNOLOGY

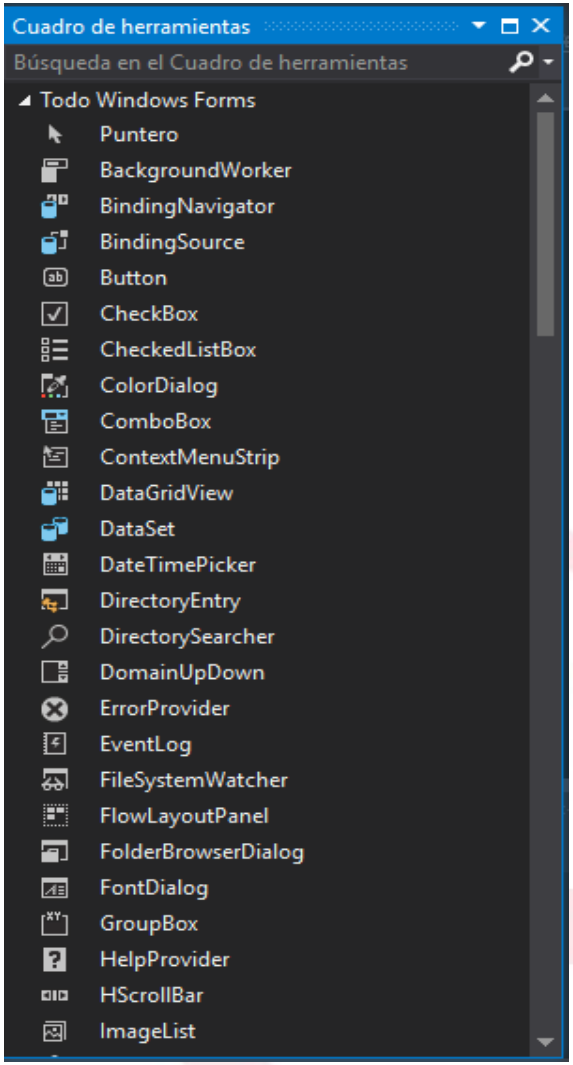
Server Explorer

- Permite crear y manipular conexiones a bases de datos
- Conectarse a servers y explorar su contenido



ToolBox

- Muestra ítems para utilizar en los distintos proyectos de Visual Studio
- Los ítems cambian dependiendo del proyecto
  - Componentes .NET
  - Componentes COM
  - Objetos HTML
  - Fragmentos de Código



HEINSOHN
BUSINESS TECHNOLOGY

Aplicación de consola con Visual Studio

- El template de proyecto mas simple
  - Agrega elementos para crear una aplicación en modo consola
  - Típicamente son las aplicaciones que se diseñan sin interfaz de usuario gráfica y se compilan en un único ejecutable
  - Se ejecutan desde la línea de comando
  - Buena herramienta para el estudio de nuevas técnicas de desarrollo de aplicaciones, sin preocuparse por la UI

Demo

- Primera Aplicación Modo Consola

Resumen

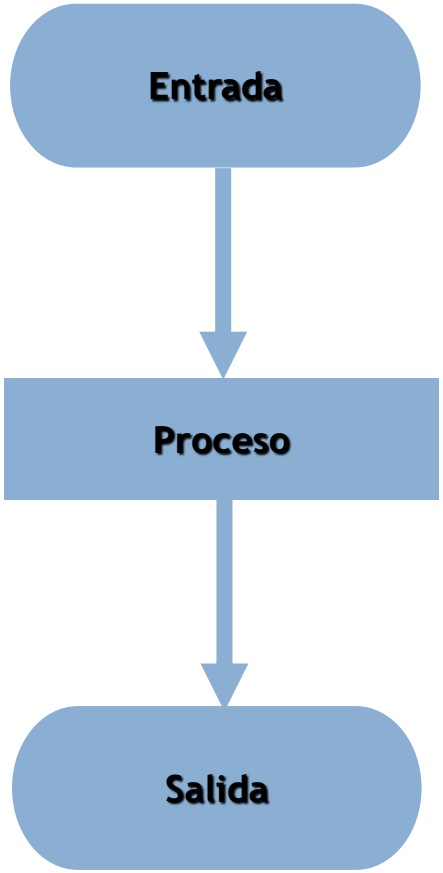
- Herramienta de desarrollo
- Proyectos y Soluciones
- Creación de aplicación en modo consola

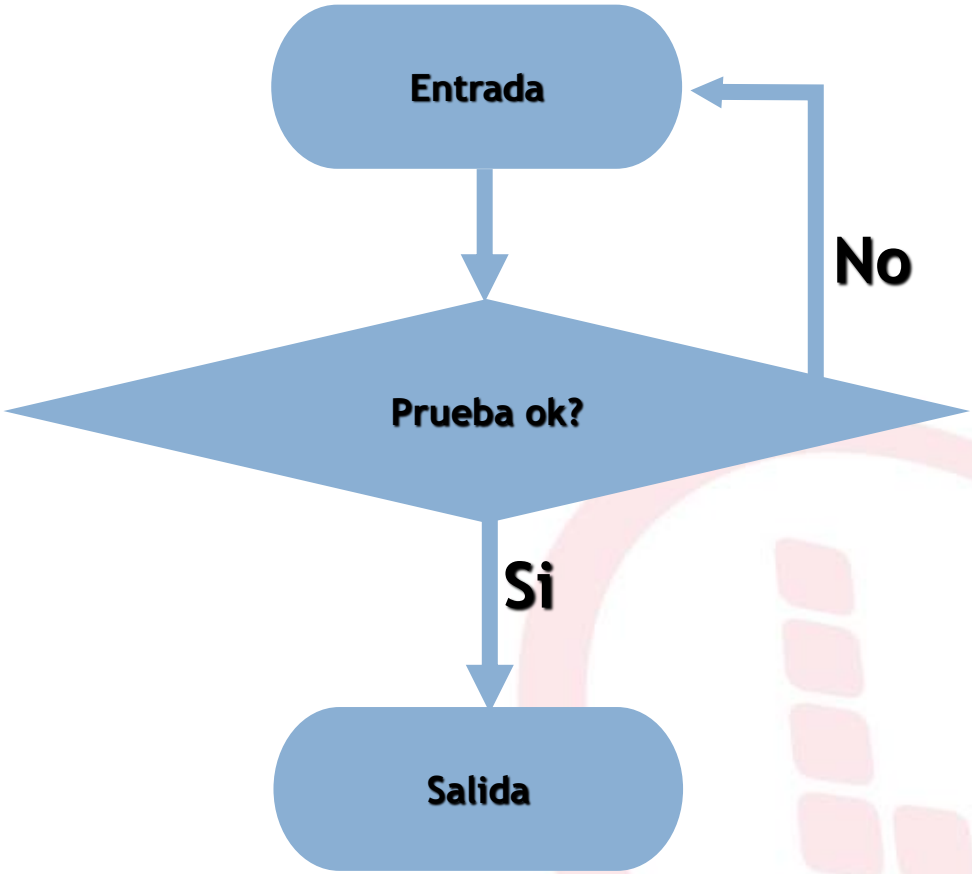
HEINSOHN
BUSINESS TECHNOLOGY

Agenda

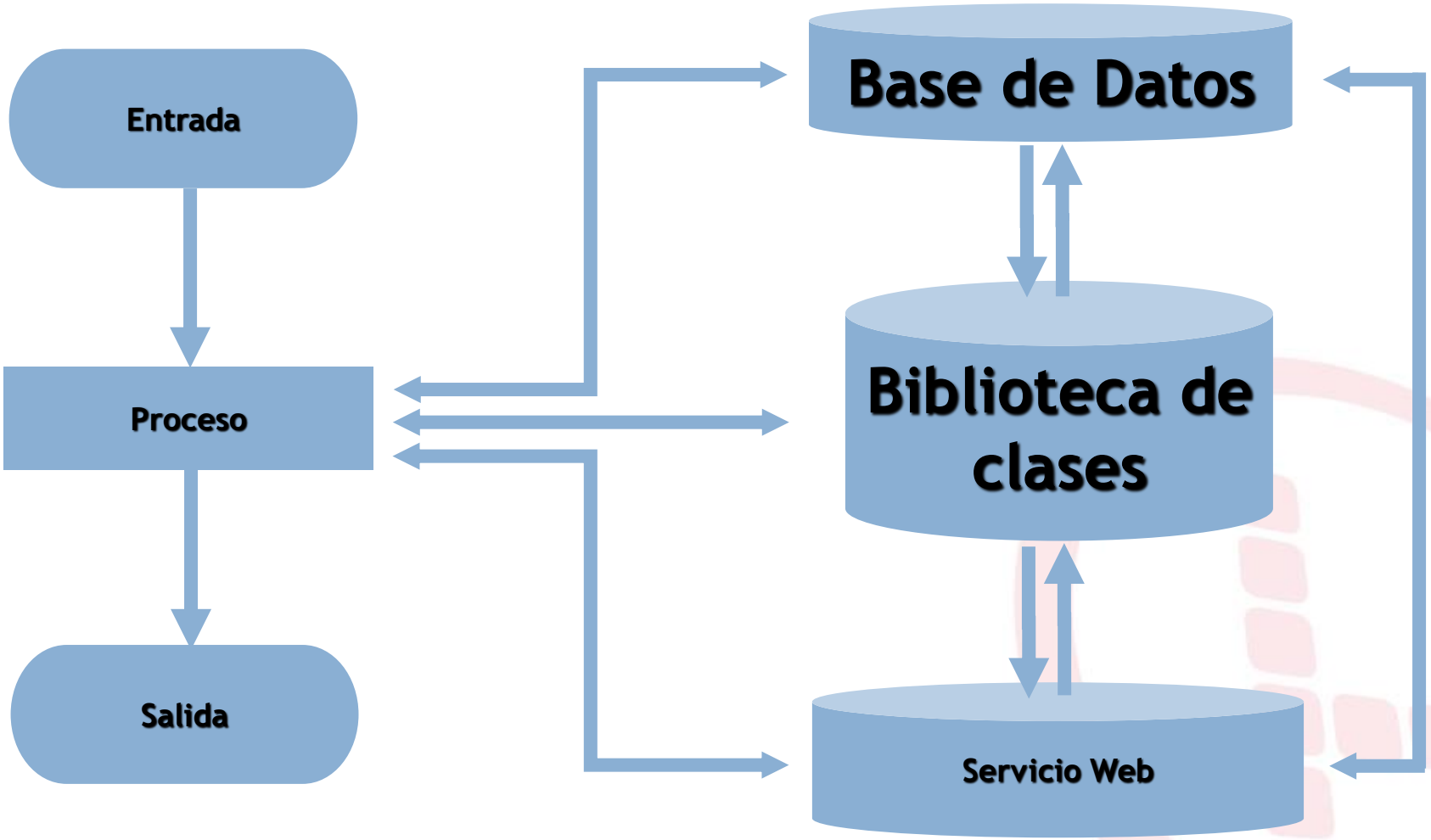
- IDE Visual Studio 2005
  - Administrando Proyectos y Soluciones
  - Herramientas
  - Creación de la primera aplicación
- Sintaxis de los lenguajes
  - Lógica de programación
  - Elección del lenguaje
  - Variables y tipos de datos
  - Estructuras lógicas

Lógica de Programación





Lógica de Programación



HEINSOHN
BUSINESS TECHNOLOGY

Visual Basic .NET y C#

VB .NET

Re-escrito desde cero para trabajar bajo .NET. Ahora totalmente OO. Mejoras s/VB6:

- Herencia
- Sobrecarga
- Constructores
- Administración estructurada de excepciones
- Comprobación de Tipos
- Miembros Shared

Visual C#

Fue un lenguaje creado especialmente para .NET. Totalmente OO.

- **Sintaxis similar a C++, J**
- **Tipos seguros**
- **Case-sensitive**
- **Atributos accedidos por un punto**
- **Todo es tratado como objetos**

Sintaxis

HEINSOHN
BUSINESS TECHNOLOGY

Elección del lenguaje

- .NET utiliza UN solo runtime (el CLR) y TODO lenguaje para .NET compila a MSIL
- Prácticamente no hay diferencias de performance entre VB.NET y C#.
- Cual lenguaje usar, en gral. dependerá de su experiencia previa con otros lenguajes o decisión personal.
  - Si conoce Java, C++, etc. >> C#
  - Si conoce VB o VBScript >> VB.NET

HEINSOHN
BUSINESS TECHNOLOGY

Variables

- ¿Qué es una variable?
- ¿En qué situación se usa una variable?
- Variables en .NET
  - Declaradas en cualquier lugar del lugar del código
  - Todas deben tener un tipo
  - El contenido de la variable tiene que estar de acuerdo con su definición

HEINSOHN
BUSINESS TECHNOLOGY

Variables - Declaración

- C#: el tipo de variable precede al identificador

```
int x;
decimal y;
rectangle z;
Cliente cli;
```

Variables - Inicialización

- C#: toda variable debe ser inicializada EXPLICITAMENTE antes de ser usada

```
int tempBalance; //variable local
//ERROR: tempBalance NO ha sido inicializada
```

HEINSOHN
BUSINESS TECHNOLOGY

Variables - Nomenclaturas

- ¿Cómo declarar una variable?
- Nomenclaturas y convenciones
  - Notación Húngara

```
//C#
int inValor;
string strTexto = "Visual C#";
long lngValor = 4294967296;
double dblSueldo = 129.90;
object Persona;
```

HEINSOHN
BUSINESS TECHNOLOGY

Variables - Tipos

Visual Basic type	Common language runtime type structure	Nominal storage allocation	Value range
Boolean	**System.Boolean**	2 bytes	**True** or **False.**
Byte	**System.Byte**	1 byte	0 through 255 (unsigned).
Char	**System.Char**	2 bytes	0 through 65535 (unsigned).
Date	**System.DateTime**	8 bytes	0:00:00 on January 1, 0001 through 11:59:59 PM on December 31, 9999.
Decimal	**System.Decimal**	16 bytes	0 through +/- 79,228,162,514,264,337,593,543,950,335 with no decimal point; 0 through +/- 7.9228162514264337593543950335 with 28 places to the right of the decimal; smallest nonzero number is +/-0.0000000000000000000000000001 (+/-1E-28).
Double (double-precision floating-point)	**System.Double**	8 bytes	-1.79769313486231570E+308 through -4.94065645841246544E-324 for negative values; 4.94065645841246544E-324 through 1.79769313486231570E+308 for positive values.

HEINSOHN
BUSINESS TECHNOLOGY

Variables - Tipos

Visual Basic type	Common language runtime type structure	Nominal storage allocation	Value range
Long (long integer)	**System.Int64**	8 bytes	-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807.
Object	**System.Object** (class)	4 bytes	Any type can be stored in a variable of type **Object**.
Short	**System.Int16**	2 bytes	-32,768 through 32,767.
Single (single-precision floating-point)	**System.Single**	4 bytes	-3.4028235E+38 through -1.401298E-45 for negative values; 1.401298E-45 through 3.4028235E+38 for positive values.
String (variable-length)	**System.String** (class)	Depends on implementing platform	0 to approximately 2 billion Unicode characters.
User-Defined Type (structure)	(inherits from **System.ValueType**)	Depends on implementing platform	Each member of the structure has a range determined by its data type and independent of the ranges of the other members.

HEINSOHN
BUSINESS TECHNOLOGY

Variables - Alcance

```
string fuera = "Declarada fuera"
string temp = ""
If ( mostrarValores )
{
      string dentro = "Mostrada Dentro"
      temp = dentro
}
else
{
      temp = fuera
}
```

Sintaxis

HEINSOHN
BUSINESS TECHNOLOGY

Case Sensitive

- C# es case-sensitivity

```
system.console.writeline("HOLA");   INCORRECTO

System.Console.WriteLine(           CORRECTO
```

HEINSOHN
BUSINESS TECHNOLOGY

Terminaciones de línea

- C# la línea finaliza con un ;

```
//Una linea con mas de un renglon
string sName = sFirstName +
               sLastName;
//El punto y coma indica FINAL de linea
```

Comentarios

- C# soporta dos tipos de comentarios

```
// Comentario de una sola linea
string sName = "Juan";
/* Comentario con mas
   de un renglon */
```

HEINSOHN
BUSINESS TECHNOLOGY

Laboratorio

- Primera aplicación de consola

Estructuras de Decisión - IF

IF - C#

```
if (condición)
   statements
else if
   (condición)
   statements
else
   statements
```

HEINSOHN
BUSINESS TECHNOLOGY

Operadores Logicos

C#	VB.NET	Operador
&&	And	**Operador logico Y**
ll	Or	**Operador logico O**
!	Not	**Negacion logica**
==	=	**Igual**
!=	<>	**Distinto**

- **En C# todas las evaluaciones se hacen por "cortocircuito"**

```
//Si Hacer1() es True, entonces
//NO se evalua Hacer2()
if  Hacer1() || Hacer2())
```

```
//Si Hacer1() es False, entonces
//NO se evalua Hacer2()
if  Hacer1() && Hacer2())
```

HEINSOHN
BUSINESS TECHNOLOGY

If Else - Ejemplo I

```
if (x > 10)
   HacerAlgo();
```

```
if (x < 10)
{
    Hacer1();
    Hacer2();
}
```

```
if (x < 10)
{
    Hacer1();
}
else
{
    Hacer2();
}
```

```
if (x < 10)
{
    Hacer1();
}
else if (x > 20)
{
    Hacer2();
}
else
{
    Hacer3();
}
```

If Else - Ejemplo II

```
if (nombre ==“Juan”)
    HacerAlgo();

if (nombre != “Carlos”)
    {
        Hacer1();
        Hacer2();
    }

if ( nombre==“Juan” ||
     nombre==“Carlos)
    {
        Hacer1();
    }
```

HEINSOHN
BUSINESS TECHNOLOGY

Laboratorio

- Estructura de decisión

HEINSOHN
BUSINESS TECHNOLOGY

Estructuras de decisión - Case

switch - C#

```
switch (expresion-a-
   evaluar) {
case valor:
   statements
    break;
default:
   statements
   break; }
```


Sintaxis

HEINSOHN
BUSINESS TECHNOLOGY

switch - C#

```
string Pais = valor;
string Deporte = "";
swktch (Pais)
        case "Brasil":
         Deporte = "Futbol";
        break;
        case "USA":
         Deporte = "Basquet";
        default:
         Deporte = "Tenis"
        break;
```

```
int opcion = valor;
string Deporte = "";
swktch (opcion)
        case 1:
         Deporte = "Futbol";
        break;
        case 2:
         Deporte = "Basquet";
        default:
         Deporte = "Tenis"
        break;
```

HEINSOHN
BUSINESS TECHNOLOGY

Laboratorio

- Estructura de decisión II

Arreglos

- C# utiliza corchetes [ ] para definición de arrays

```
string                  //Definicion de un Arreglo de strings
telefonos = new string[3]; //De 3 elementos
telefonos[0] = "1245"; //Seteo del 1er elemento del arreglo

//Definicion y asignacion de una vez
telefonos = new string[] {"1","2","3"};
```

HEINSOHN
BUSINESS TECHNOLOGY

Estructuras de iteración - For

- **C#: la sentencia for consta de tres partes**

```
//Partes: declaración, prueba, acción
for (int i=1; i < 10; i++)
{
}
```

HEINSOHN
BUSINESS TECHNOLOGY

For - C#

for (contador; expresion; incremento)
{
statements
}

```
for (int i = 1; i<=10; i++)
        Console.WriteLine(i);
```

```
for (int i = 1; i<=10; i++)
{
        Console.WriteLine(i);
}
```

Sintaxis

HEINSOHN
BUSINESS TECHNOLOGY

Laboratorio

- Estructura de Iteración

HEINSOHN
BUSINESS TECHNOLOGY

Estructuras de Iteración - for / each

- For/Each permite recorrer arreglos y colecciones
- C#: usa la palabra foreach

```
string[] nombres = new string[5];
foreach(string auxNombre in nombres)
{
    //auxNombre es de SOLO LECTURA
}
```

foreach - C#

foreach (elemento in grupo){
statements }

```
int multDos = 0;
int noMult = 0;
int[] arrayData = {1, 5, 8, 45, 25};
foreach (int numero In arrayData)
{
        if (numero MOD 2 ==  0)
                multDos += 1;
        else
                noMult +=1;
}
```

HEINSOHN
BUSINESS TECHNOLOGY

Estructuras de Iteración - While

- C#: usa las palabras while o do - while

```
bool condicion = true;
while (condicion)
{
    //codigo que haga que cambie la condicion
}
```

HEINSOHN
BUSINESS TECHNOLOGY

While - Todas las opciones

C#

```
while (expresion){ statements }
do{ statements } while
  (expresion);
```

HEINSOHN
BUSINESS TECHNOLOGY

Excepciones

- Ejecución de un programa en forma irregular
- Código bien escrito debe controlarlas
- .NET cuenta con gran soporte para el manejo de excepciones

HEINSOHN
BUSINESS TECHNOLOGY

Excepciones en C# y VB.NET

- Excepción: objeto que se genera cuando en tiempo de ejecución ocurre un error y contiene info sobre el mismo

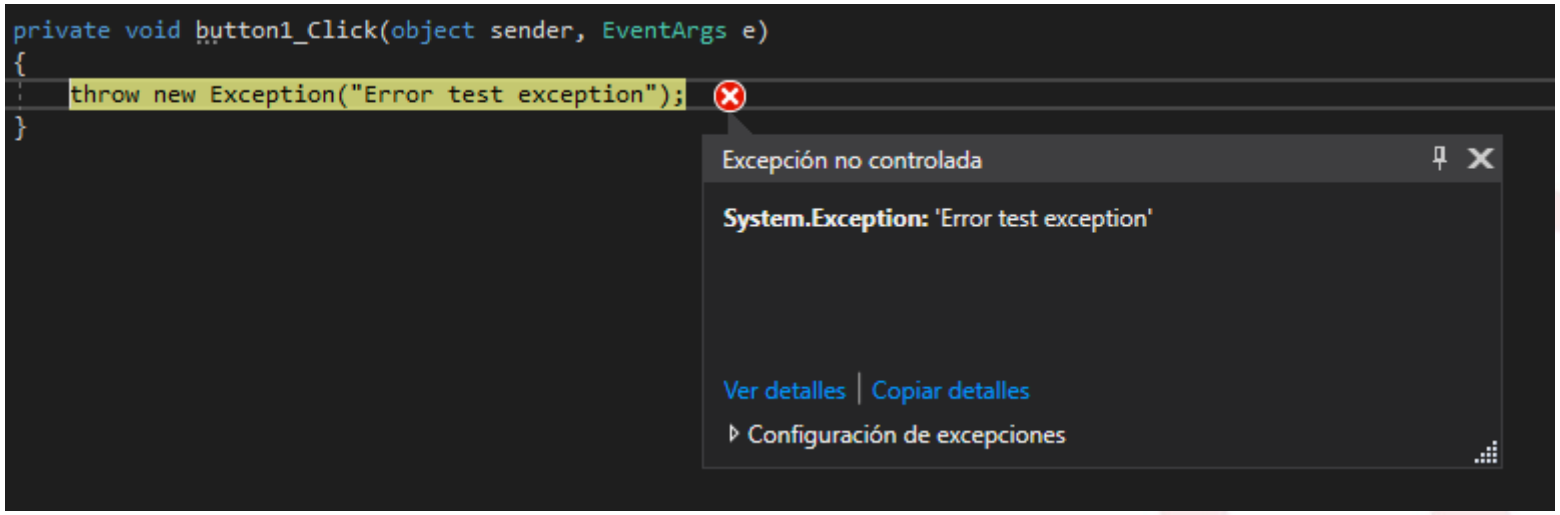
- C#: usa las palabras try/catch/finally

```
try
{
    int resultado = x/y;
}
catch(DivideByZeroException e)
{
    //Error division por cero
}
catch
{
    //Otro error
}
finally
{
    //Siempre pasa por aca
}
```

HEINSOHN
BUSINESS TECHNOLOGY

Asistente manejo de Excepciones

- Permite Descubrir mas sobre una excepción.
- Permite corregir errores en Run-Time



HEINSOHN
BUSINESS TECHNOLOGY

Resumen

- Lógica de programación
- Elección del lenguaje
- Variables
- Estructuras de decisión e iteración
- Excepciones

HEINSOHN
BUSINESS TECHNOLOGY