

Solución a prueba Técnica:

Nombre: Fabián López Quesada.

Cédula: 1-1523-0265.

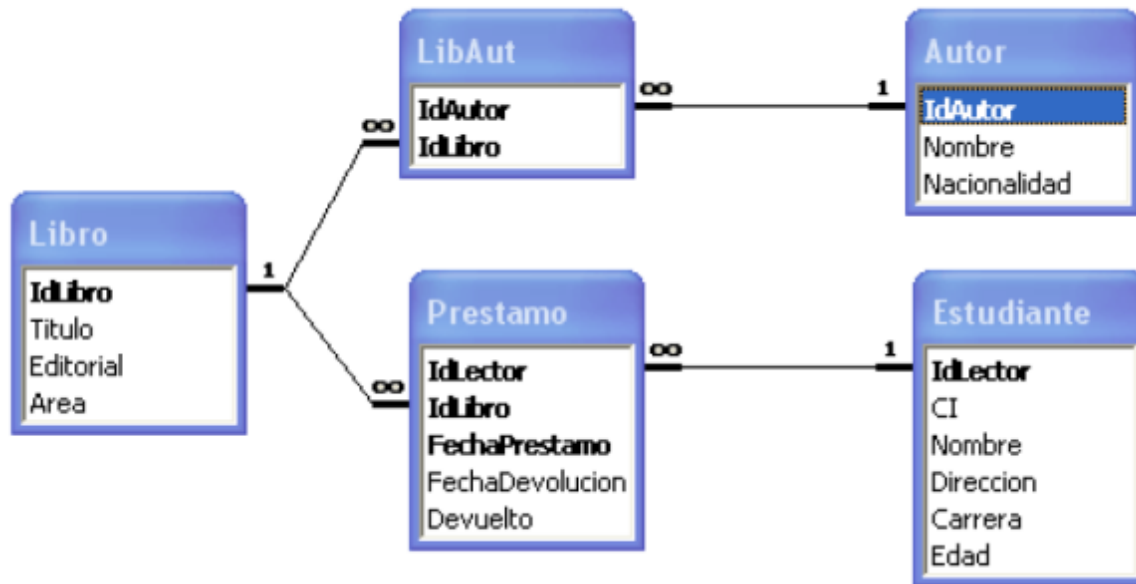
Correo: fabianlq1093@gmail.com

Teléfono: 506 – 6475 0542.

Prueba Técnica Enlace

Comprobación de conocimiento SQL

Dada la siguiente estructura de base de datos, diseñe las consultas SQL a las preguntas planteadas



Script de creación de la base de datos, para ello se utiliza Microsoft SQL Server 2016:

```
CREATE DATABASE LIBRARY
GO
```

```
USE LIBRARY
GO
```

```
CREATE TABLE Libro
(
    IdLibro int primary key not null identity(1,1),
    Titulo NVARCHAR(200) NOT NULL,
    Editorial NVARCHAR(200) NOT NULL,
    Area NVARCHAR(200) NOT NULL
)
```

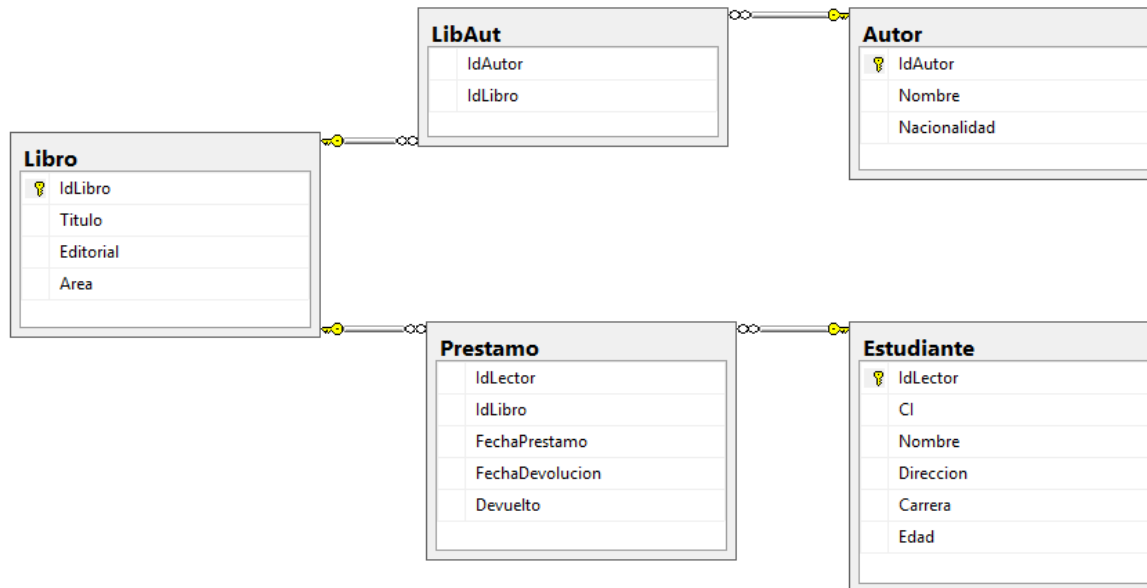
```
CREATE TABLE Autor
(
    IdAutor int primary key not null identity(1,1),
    Nombre NVARCHAR(200) NOT NULL,
    Nacionalidad NVARCHAR(200) NOT NULL
)
```

```
CREATE TABLE Estudiante
(
    IdLector int primary key not null identity(1,1),
    CI NVARCHAR(200) NOT NULL,
    Nombre NVARCHAR(200) NOT NULL,
    Direccion NVARCHAR(200) NOT NULL,
    Carrera NVARCHAR(200) NOT NULL,
    Edad INT NOT NULL
)
```

```
CREATE TABLE LibAut
(
    IdAutor int not null,
    IdLibro int not null,
    CONSTRAINT FK_IdAutor FOREIGN KEY (IdAutor)
        REFERENCES Autor (IdAutor),
    CONSTRAINT FK_IdLibro FOREIGN KEY (IdLibro)
        REFERENCES Libro (IdLibro)
)
```

```
CREATE TABLE Prestamo
(
    IdLector int not null,
    IdLibro int not null,
    FechaPrestamo date not null,
    FechaDevolucion date not null,
    Devuelto bit,
    CONSTRAINT FK_IdLector FOREIGN KEY (IdLector)
        REFERENCES Estudiante (IdLector),
    CONSTRAINT FK_IdLibros FOREIGN KEY (IdLibro)
        REFERENCES Libro (IdLibro)
)
```

Diagrama generado desde el SQL Server Management Studio:



1. Listar los nombres de los estudiantes cuyo apellido comience con la letra G.

Datos insertados en la tabla:

IdLector	CI	Nombre	Direccion	Carrera	Edad
1	115230265	Fabian Lopez Quesada	Cartago	Computacion	30
2	113330827	Jennifer Valverde Ramirez	San Jose	Arquitectura	35
3	305180531	Ivette Jimenez Quiros	Alajuela	Biologia	48
5	604367812	Sergio Gutierrez Calderon	Puntarenas	Derecho	37
6	602431352	Gabrial Garcia Morales	Limon	Odontologia	44

En este ejercicio, se asume por parte de mi persona que la columna “NOMBRE” contiene todo el nombre completo compuesto por:

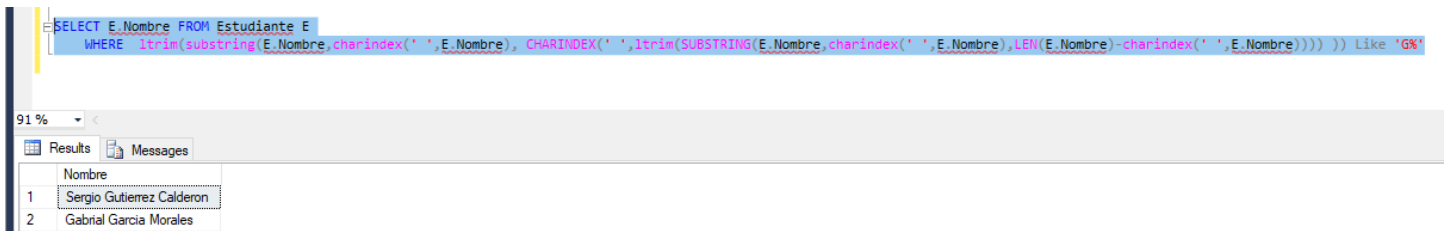
[PRIMER NOMBRE] + [APELLIDO 1] + [APELLIDO 2]

En este caso la solución sería:

```
SELECT E.Nombre FROM Estudiante E
WHERE ltrim(substring(E.Nombre,charindex(' ',E.Nombre), CHARINDEX(' ',ltrim(SUBSTRING(E.Nombre,charindex(' ',E.Nombre),LEN(E.Nombre)-charindex(' ',E.Nombre)))))) Like 'L%'
```

Lo que se está haciendo, es separando todo el contenido de la columna para cada elemento de la tabla, y tomando solamente la segunda palabra, utilizando la sentencia LIKE donde solo comience por la letra “G”

Resultado de la consulta:



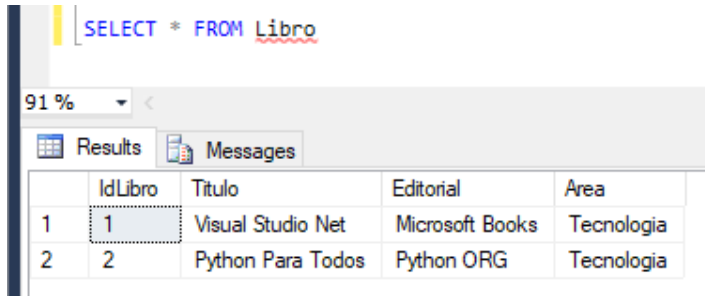
Nombre
1 Sergio Gutierrez Calderon
2 Gabrial Garcia Morales

NOTA: Asumí esto ya que la tabla NO contiene una estructura de nombre donde se muestre una columna para primer nombre, segundo nombre, primer apellido y segundo apellido. Si esta estructura existiera, entonces la consulta que habría que realizar sería la siguiente:

```
SELECT E.Nombre FROM Estudiante E
WHERE E.PrimerApellido Like 'G%'
```

2. ¿Quiénes son los autores del libro “Visual Studio Net”? Listar solamente los nombres.

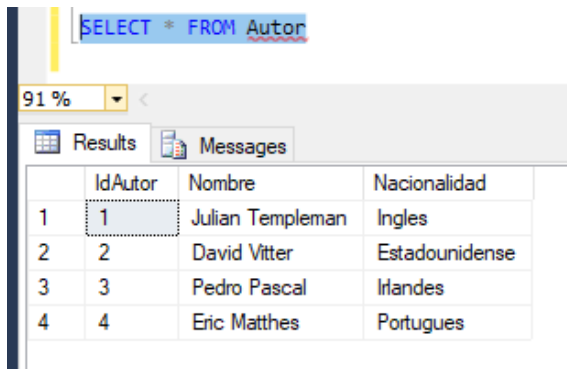
En este caso, se agrega un par de registros a la tabla de Libros:



The screenshot shows a SQL query window with the command `SELECT * FROM Libro`. Below the query, the 'Results' tab displays a table with the following data:

	IdLibro	Titulo	Editorial	Area
1	1	Visual Studio Net	Microsoft Books	Tecnologia
2	2	Python Para Todos	Python ORG	Tecnologia

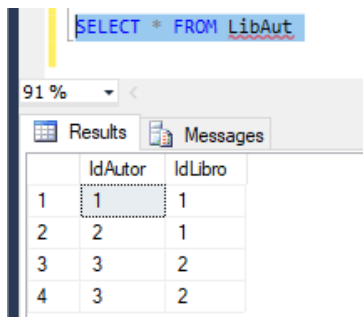
Se va a crear en la tabla Autor, 4 autores distintos de libros.



The screenshot shows a SQL query window with the command `SELECT * FROM Autor`. Below the query, the 'Results' tab displays a table with the following data:

	IdAutor	Nombre	Nacionalidad
1	1	Julian Templeman	Ingles
2	2	David Vitter	Estadounidense
3	3	Pedro Pascal	Irlandes
4	4	Eric Matthes	Portugues

Adicional a esto, se va a agregar a la tabla LibAut para la prueba, 2 autores para cada libro para poder representar el ejemplo.

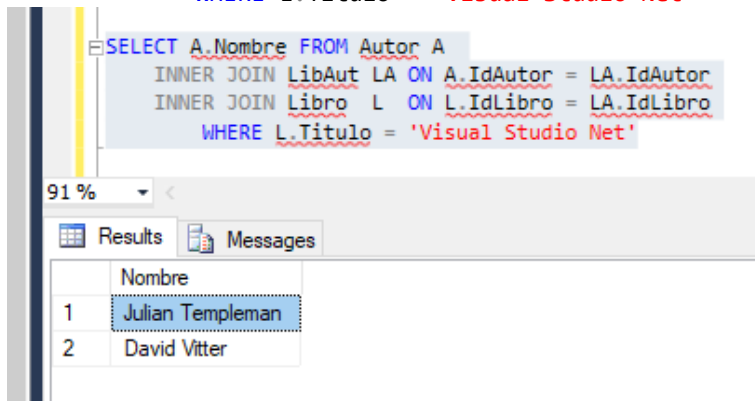


The screenshot shows a SQL query window with the command `SELECT * FROM LibAut`. Below the query, the 'Results' tab displays a table with the following data:

	IdAutor	IdLibro
1	1	1
2	2	1
3	3	2
4	3	2

El resultado de la consulta sería:

```
SELECT A.Nombre FROM Autor A
INNER JOIN LibAut LA ON A.IdAutor = LA.IdAutor
INNER JOIN Libro L ON L.IdLibro = LA.IdLibro
WHERE L.Titulo = 'Visual Studio Net'
```



The screenshot shows the same SQL query window with the query executed. The 'Results' tab displays the following output:

	Nombre
1	Julian Templeman
2	David Vitter

3. Listar los nombres de los estudiantes a los que se prestaron libros del área de “Base de Datos”.

En este caso, se crean dos libros en la tabla “Libros” que contienen el área “Base de Datos”:

```
SELECT * FROM Libro
```

	IdLibro	Titulo	Editorial	Area
1	1	Visual Studio Net	Microsoft Books	Tecnologia
2	2	Python Para Todos	Python ORG	Tecnologia
3	3	SQL Microsoft	Microsoft Books	Base de Datos
4	4	MySQL	Sun MS	Base de Datos

Se va a agregar un conjunto de Prestamos de estudiantes y libros, incluyendo libros que pertenecen al área “Base de Datos”.

```
SELECT * FROM Prestamo
```

	IdLector	IdLibro	FechaPrestamo	FechaDevolucion	Devuelto
1	1	1	2007-03-10	2007-04-10	0
2	2	2	2007-03-10	2007-05-21	0
3	3	4	2022-01-01	2022-03-04	1
4	3	3	2021-03-12	2022-03-29	1

El resultado sería el siguiente:

```
SELECT DISTINCT E.Nombre FROM Estudiante E  
INNER JOIN Prestamo P ON E.IdLector = P.IdLector  
INNER JOIN Libro L ON L.IdLibro = P.IdLibro  
WHERE L.Area = 'Base de Datos'
```

	Nombre
1	Ivette Jimenez Quiros

Se utiliza la sentencia DISTINCT debido a que podría existir más de una vez el mismo nombre, por lo que se excluyen los repetidos para que no afecte a la hora de visualizar la información. En este caso, se le ha prestado 2 veces libros con el área Base de Datos a la misma persona, por esa razón, la consulta retornaba 2 registros, al utilizar el Distinct, ya se visualiza de manera correcta.

4. Listar los títulos de los libros que debían devolverse el “10/04/2007”.

Se utilizará los registros insertados previamente en la tabla “Estudiante”

```
SELECT * FROM Estudiante
```

	IdLector	CI	Nombre	Direccion	Camera	Edad
1	1	115230265	Fabian Lopez Quesada	Cartago	Computacion	30
2	2	113330827	Jennifer Valverde Ramirez	San Jose	Arquitectura	35
3	3	305180531	Ivette Jimenez Quiros	Alajuela	Biologia	48
4	5	604367812	Sergio Gutierrez Calderon	Puntarenas	Derecho	37
5	6	602431352	Gabrial Garcia Morales	Limon	Odontologia	44

Se utiliza los datos previamente insertados en la tabla Prestamo, donde se puede visualizar que existe 1 libro que debía devolverse el 10/04/2007.

```
SELECT * FROM Prestamo
```

	IdLector	IdLibro	FechaPrestamo	FechaDevolucion	Devuelto
1	1	1	2007-03-10	2007-04-10	0
2	2	2	2007-03-10	2007-05-21	0
3	3	4	2022-01-01	2022-03-04	1
4	3	3	2021-03-12	2022-03-29	1

El resultado de la consulta sería solo 1 registro, ya que solamente un registro posee esa fecha:

```
SELECT L.Titulo FROM Libro L
INNER JOIN Prestamo P ON L.IdLibro = P.IdLibro
WHERE P.FechaDevolucion = '2007-04-10'
```

```
SELECT L.Titulo FROM Libro L
INNER JOIN Prestamo P ON L.IdLibro = P.IdLibro
WHERE P.FechaDevolucion = '2007-04-10'
```

	Titulo
1	Visual Studio Net

5. Listar los datos de los estudiantes cuya edad es mayor al promedio.

En este caso, se utiliza la función AVG para poder obtener el promedio, con los registros que existen en la tabla, la edad promedio es 38 años.

	IdLector	CI	Nombre	Direccion	Carrera	Edad
1	1	115230265	Fabian Lopez Quesada	Cartago	Computacion	30
2	2	113330827	Jennifer Valverde Ramirez	San Jose	Arquitectura	35
3	3	305180531	Ivette Jimenez Quiros	Alajuela	Biologia	48
4	5	604367812	Sergio Gutierrez Calderon	Puntarenas	Derecho	37
5	6	602431352	Gabrial Garcia Morales	Limon	Odontologia	44

El resultado de la consulta sería:

```
SELECT E.IdLector, E.CI, E.Nombre, E.Direccion, E.Carrera, E.Edad, (SELECT AVG(C.Edad) FROM Estudiante C) AS EdadPromedio
FROM ESTUDIANTE E
WHERE E.Edad > (SELECT AVG(C.Edad) FROM Estudiante C)
```

91 %

Results Messages

	IdLector	CI	Nombre	Direccion	Carrera	Edad	EdadPromedio
1	3	305180531	Ivette Jimenez Quiros	Alajuela	Biologia	48	38
2	6	602431352	Gabrial Garcia Morales	Limon	Odontologia	44	38

6. Agrupar por título y editorial y a su vez mostrar la cantidad de estudiantes que posean préstamos de libros sin devolver.

Para poder obtener este dato, hay que agrupar los datos, a continuación de muestra a nivel general cómo se encuentran los préstamos, donde la columna “Devuelto” posee un valor de tipo BIT, siendo 0 que NO se ha devuelto, y 1 que Si.

	Titulo	Editorial	idLector	Nombre	Devuelto
1	Visual Studio Net	Microsoft Books	1	Fabian Lopez Quesada	0
2	Python Para Todos	Python ORG	2	Jennifer Valverde Ramirez	0
3	MySQL	Sun MS	3	Ivette Jimenez Quiros	1
4	SQL Microsoft	Microsoft Books	3	Ivette Jimenez Quiros	1

La solución a esta pregunta sería:

```
SELECT L.Titulo, L.Editorial, COUNT(E.idLector) AS CantidadEstudiantes FROM Libro L
INNER JOIN Prestamo P ON P.idLibro = L.idLibro
INNER JOIN Estudiante E ON P.idLector = E.idLector
WHERE P.Devuelto = 0
GROUP BY
    L.Titulo, L.Editorial
```



```

SELECT L.Titulo, L.Editorial, COUNT(E.idLector) AS CantidadEstudiantes FROM Libro L
INNER JOIN Prestamo P ON P.idLibro = L.idLibro
INNER JOIN Estudiante E ON P.idLector = E.idLector
WHERE P.Devuelto = 0
GROUP BY
L.Titulo, L.Editorial

```

91 %

Results Messages

	Titulo	Editorial	CantidadEstudiantes
1	Visual Studio Net	Microsoft Books	1
2	Python Para Todos	Python ORG	1

7. Listar los estudiantes que posean préstamos sin devolver de libros que tengan al menos un autor de nacionalidad “Costarricense”.

Se va a agregar un autor más con nacionalidad “Costarricense” a la tabla de Autor:

	IdAutor	Nombre	Nacionalidad
1	1	Julian Templeman	Ingles
2	2	David Vitter	Estadounidense
3	3	Pedro Pascal	Irlandes
4	4	Eric Matthes	Portugues
5	5	Fabian Lopez	Costarricense

Posterior, se va a hacer una relación en la tabla LibAut para indicar que el libro Visual Studio NET también pertenece al autor Fabian Lopez.

	IdAutor	IdLibro
1	1	1
2	2	1
3	3	2
4	4	2
5	5	1

Se debe hacer relación entre las tablas Prestamo, Estudiante, LibAut y Autor para poder obtener la consulta, adicional, se utiliza la palabra DISTINCT para no repetir nombres de estudiantes que cumplan con esta condición.

A continuación se muestra cómo están los datos de prestamos sin devolver por estudiante, autor y nacionalidad.

	Titulo	Editorial	idLector	NombreEstudiante	Devuelto	NombreAutor	Nacionalidad
1	Visual Studio Net	Microsoft Books	1	Fabian Lopez Quesada	0	Julian Templeman	Ingles
2	Visual Studio Net	Microsoft Books	1	Fabian Lopez Quesada	0	David Vitter	Estadounidense
3	Python Para Todos	Python ORG	2	Jennifer Valverde Ramirez	0	Pedro Pascal	Irlandes
4	Python Para Todos	Python ORG	2	Jennifer Valverde Ramirez	0	Eric Matthes	Portugues
5	Visual Studio Net	Microsoft Books	1	Fabian Lopez Quesada	0	Fabian Lopez	Costarricense

La solución sería la siguiente:

```

SELECT DISTINCT E.Nombre AS NombreEstudiante
FROM Libro L

```

```
INNER JOIN Prestamo P      ON P.idLibro = L.idLibro
INNER JOIN Estudiante E    ON P.idLector = E.idLector
INNER JOIN LibAut LA       ON LA.IdLibro = L.IdLibro
INNER JOIN Autor A         ON LA.idautor = A.idAutor
WHERE
    P.Devuelto = 0
    AND A.Nacionalidad = 'Costarricense'
```

```
SELECT DISTINCT E.Nombre AS NombreEstudiante
FROM Libro L
INNER JOIN Prestamo P ON P.idLibro = L.idLibro
INNER JOIN Estudiante E ON P.idLector = E.idLector
INNER JOIN LibAut LA ON LA.IdLibro = L.IdLibro
INNER JOIN Autor A ON LA.idautor = A.idAutor
WHERE
    P.Devuelto = 0
    AND A.Nacionalidad = 'Costarricense'
```

91 %

Results Messages

	NombreEstudiante
1	Fabian Lopez Quesada

Comprobación de conocimiento JavaScript

Dada la siguiente estructura HTML, responda las consultas en JavaScript.

```
<p id="counter"></p>

<table>
  <tr>
    <th>Name</th>
  </tr>
  <tbody id="countries">
  </tbody>
</table>
```

- 1- Crear un arreglo con los siguientes países 'France', 'Germany', 'England', 'Spain', 'Belgium', 'Italy', 'Portugal', 'Ireland', 'Luxembourg'

```
//Se crea el arreglo *Ejercicio 1*
```

```
lista = ['France', 'Germany', 'England', 'Spain', 'Belgium', 'Italy', 'Portugal', 'Ireland', 'Luxembourg']
```

- 2- Recorrer dicho arreglo y agregar el siguiente elemento Pais a "countries" para cada uno de los registros del arreglo.

```
//Se crea el arreglo *Ejercicio 1*
```

```
lista = ['France', 'Germany', 'England', 'Spain', 'Belgium', 'Italy', 'Portugal', 'Ireland', 'Luxembourg']
```

```
//Se llama cada elemento de la lista y se agrega al elemento countries *Ejercicio 2*
```

```
lista.forEach(myFunction);
```

```
function myFunction(item, index) {
  var table = document.getElementById("countries");
  var row = table.insertRow(0);
  var cell1 = row.insertCell(0);
  cell1.innerHTML = item;
}
```

- 3- Modificar el elemento 'Germany' y cámbielo por 'Alemania'

```
//Se sustituye Germany por la palabra Alemania *Ejercicio 3*
```

```
lista.splice(lista.indexOf("Germany"),1,"Alemania")
```

- 4- Elimine el elemento 'Spain'

```
//Se elimina del arreglo el pais Spain *Ejercicio 4*
```

```
lista.splice(lista.indexOf("Spain"),1)
```

Solución General del ejercicio:

```
<!DOCTYPE html>
<html>
<style>
table, td {
    border: 1px solid black;
}
</style>
<body>
<table>
    <tr>
        <th>Name</th>
    <tr>
        <tbody id= "countries">
        </tbody>
</table>
<script>
//Se crea el arreglo *Ejercicio 1*
lista = ['France', 'Germany', 'England','Spain', 'Belgium', 'Italy', 'Portugal', 'Ireland', 'Luxembourg']
//Se sustituye Germany por la palabra Alemania *Ejercicio 3*
lista.splice(lista.indexOf("Germany"),1,"Alemania")
//Se elimina del arreglo el pais Spain *Ejercicio 4*
lista.splice(lista.indexOf("Spain"),1)
//Se llama cada elemento de la lista y se agrega al elemento countries *Ejercicio 2*
lista.forEach(myFunction);
function myFunction(item, index) {
    var table = document.getElementById("countries");
    var row = table.insertRow(0);
    var cell1 = row.insertCell(0);
    cell1.innerHTML = item;
}
</script>
</body>
</html>
```

Resultado:

```
<!DOCTYPE html>
<html>
<style>
table, td {
  border: 1px solid black;
}
</style>
<body>
<table>
  <tr>
    <th>Name</th>
  <tr>
    <tbody id= "countries">
      </tbody>
    </table>
</script>
//Se crea el arreglo *Ejercicio 1*
lista = ['France', 'Germany', 'England','Spain', 'Belgium', 'Italy', 'Portugal', 'Irland', 'Luxembourg']
//Se sustituye Germany por la palabra Alemania *Ejercicio 3*
lista.splice(lista.indexOf("Germany"),1,"Alemania")
//Se elimina del arreglo el pais Spain *Ejercicio 4*
lista.splice(lista.indexOf("Spain"),1)
//Se llama cada elemento de la lista y se agrega al elemento countries *Ejercicio 2*
lista.forEach(myFunction);
function myFunction(item, index) {
  var table = document.getElementById("countries");
  var row = table.insertRow(0);
  var cell1 = row.insertCell(0);
  cell1.innerHTML = item;
}
</script>
</body>
</html>
```

Name
Luxembourg
Irland
Portugal
Italy
Belgium
England
Alemania
France

Comprobación de conocimiento .Net

1. ¿Según el código, que salida se obtendría y de qué tipo?

```
class Test
{
    static void Main()
    {
        double x = 1243.73;
        int c;
        int a;
        a = (int) x;
        c = 1234.73;
        System.Console.WriteLine(a);
    }
}
```

- a) «1243.73» de tipo int.
- b) «1243» de tipo int.
- c) «1234» de tipo double.
- d) «1243.73» de tipo double.

La respuesta correcta es la b.

2. Qué significa este ejemplo «If (test1)&&(test2)» en relación al operador «&&»?

- a) Solo evalúa test2 si test1 es cierto.
- b) Cierto si ambos son ciertos.
- c) Cierto si alguno de los dos es cierto.
- d) Ninguna respuesta es correcta.

La respuesta correcta es la b.

3. La visibilidad de un procedimiento viene determinada por la declaración:

- a) Private, public o internal.
- b) Return
- c) Sólo public.
- d) Todas son falsas.

La respuesta correcta es la a, pero hay que tomar en cuenta que existen también otros tipos de declaraciones que determinan su visibilidad, como protected, protected internal y private protected.

4. ¿Qué significa VOID cuando se encuentra en un método?

- a) Un método es un modelo que define a los objetos, su esqueleto y la diferencia con el objeto, es que el objeto es único, es una instancia de la clase.
- b) Significa que no va tener un valor de retorno
- c) Ayuda a mejorar y garantizar la interoperabilidad entre lenguajes mediante la definición de un conjunto de características en las que se pueden basar los programadores y que están disponibles en una gran variedad de lenguajes.
- d) Ninguna de las anteriores

La respuesta correcta es la b, al utilizar VOID en un método le estamos indicando que este NO va a retornar un valor.

5. La palabra this hace referencia a la instancia actual de la clase

- a) Verdadero
- b) Falso

La respuesta correcta es la a.

¿Qué es un constructor?

Un constructor es un método que se utiliza para inicializar un objeto. Este se crea cuando se instancia una clase. Un constructor tiene el mismo nombre que la clase y no retornan ningún valor y son utilizados para iniciar valores de campos y propiedades de un objeto cuando se crea.

¿Qué es y de un ejemplo de una sobrecarga de un método?

Sobrecarga de un método se refiere a la manera en la que se pueden definir varios métodos a la vez con el mismo nombre en una clase, pero, que a estos poseen diferentes parámetros de entrada. Es decir, esto permite que se pueda llamar al método con los parámetros según estén declarados en la clase, y se ejecutará el método que corresponda según los parámetros que son enviados.

A nivel de ejemplo, podemos definir una operación resta llamado RESTAR, que recibe dos parámetros de tipo ENTERO, y otro método que también se llama RESTAR, que recibe dos parámetros de tipo DECIMAL.

Al momento de llamar a este método, si se envían dos parámetros de tipo entero, va a invocar el primer método, de lo contrario, si se llama este método con dos parámetros de tipo DECIMAL, ejecutaría el que recibe este tipo de parámetros.