2019

Netcoreconf

Hot Crazy C#

Fernando Escolar
Developerro - Tokiota
@fernandoescolar

David Gonzalo
Developer - Tokiota
@dagope

#netcoreconf

Sponsors

Microsoft

avanade

plain concepts

iTEQUIA

encamina
PIENSA EN COLORES

東京 TOKIOTA

devs dna

3ASIDE Consultors

intelequia
SOFTWARE SOLUTIONS

minsait
An Indra company

Insight

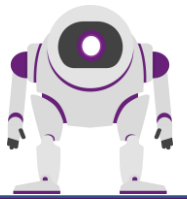
everis
an NTT DATA Company

Sothis

IRON HACK
BARCELONA

GIN MG
CASA FUNDADA
EN 1835

steema
software

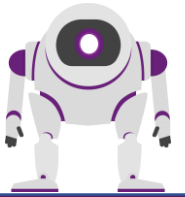



18
www.pegi.info
DISCRIMINATION
DRUGS
@*!
BAD LANGUAGE
FEAR
VIOLENCE
SEX

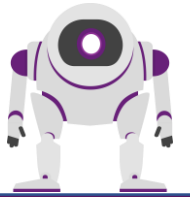
IBM

Microsoft

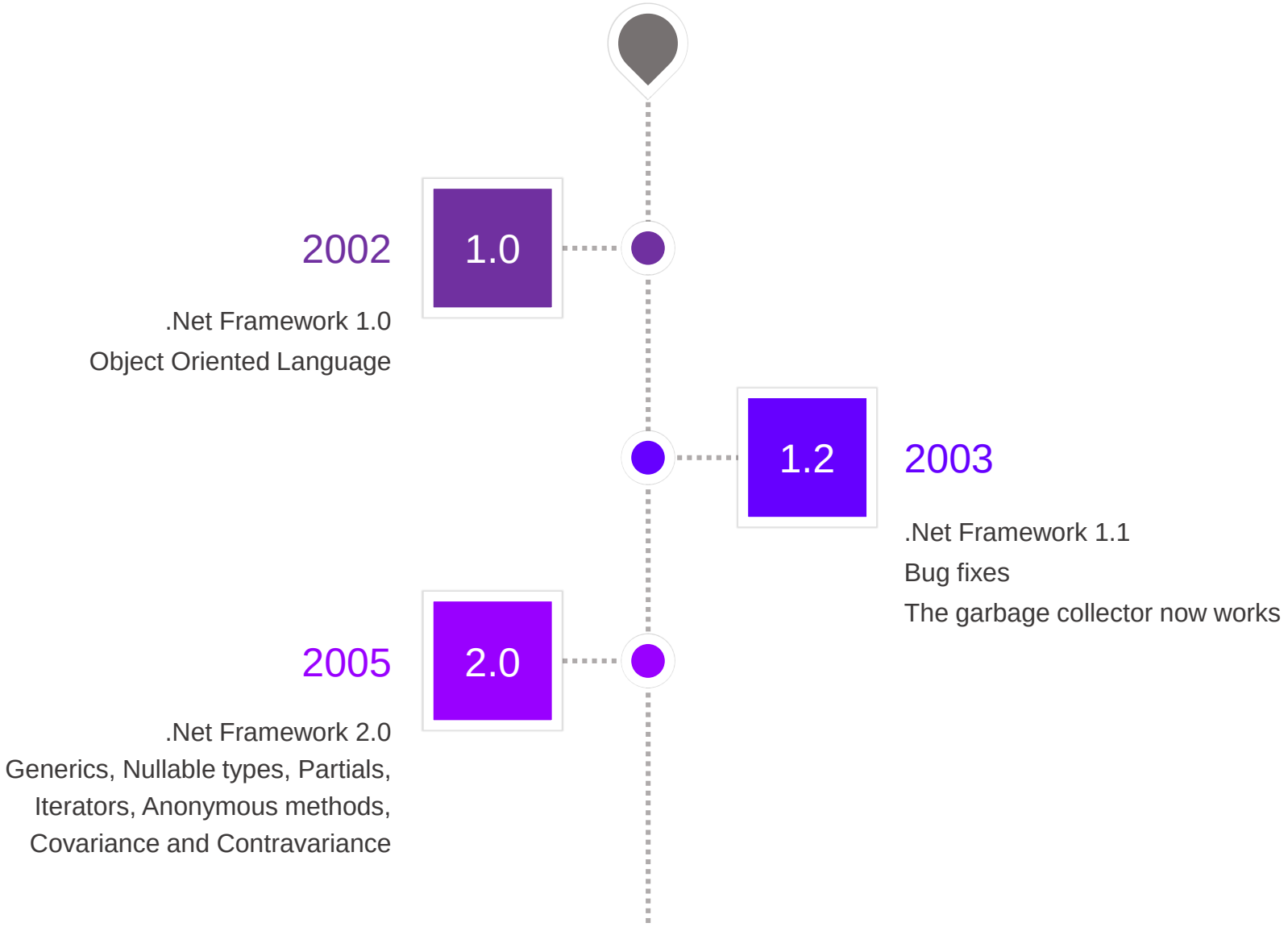
Borland

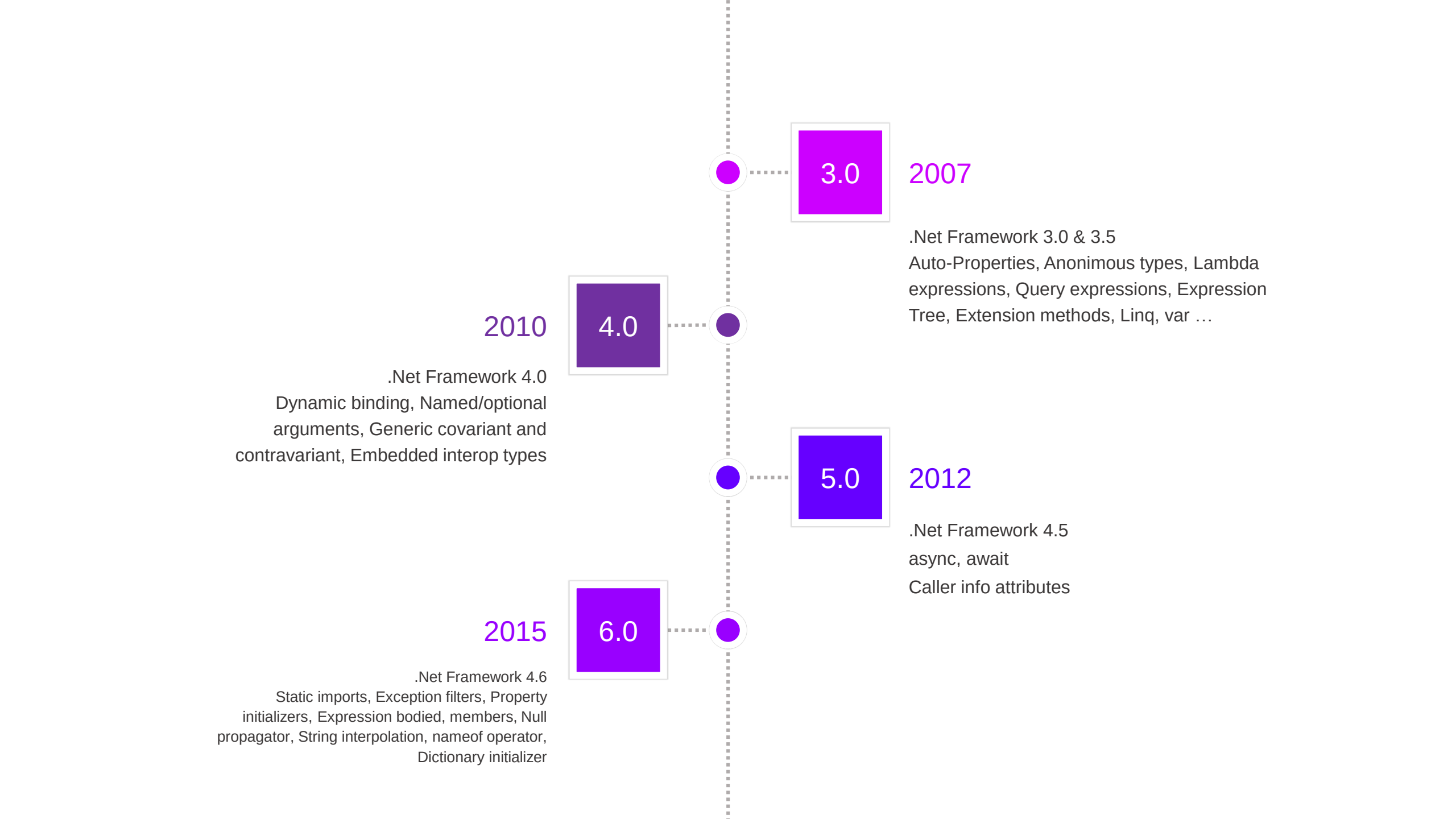


C  ++



C# Evolution


2002
1.0
.Net Framework 1.0
Object Oriented Language
1.2
2003
.Net Framework 1.1
Bug fixes
The garbage collector now works
2005
2.0
.Net Framework 2.0
Generics, Nullable types, Partials,
Iterators, Anonymous methods,
Covariance and Contravariance


3.0
2007
.Net Framework 3.0 & 3.5
Auto-Properties, Anonimous types, Lambda expressions, Query expressions, Expression Tree, Extension methods, Linq, var …
4.0
2010
.Net Framework 4.0
Dynamic binding, Named/optional arguments, Generic covariant and contravariant, Embedded interop types
5.0
2012
.Net Framework 4.5
async, await
Caller info attributes
6.0
2015
.Net Framework 4.6
Static imports, Exception filters, Property initializers, Expression bodied, members, Null propagator, String interpolation, nameof operator, Dictionary initializer

7.0

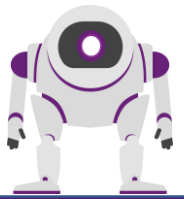
2017

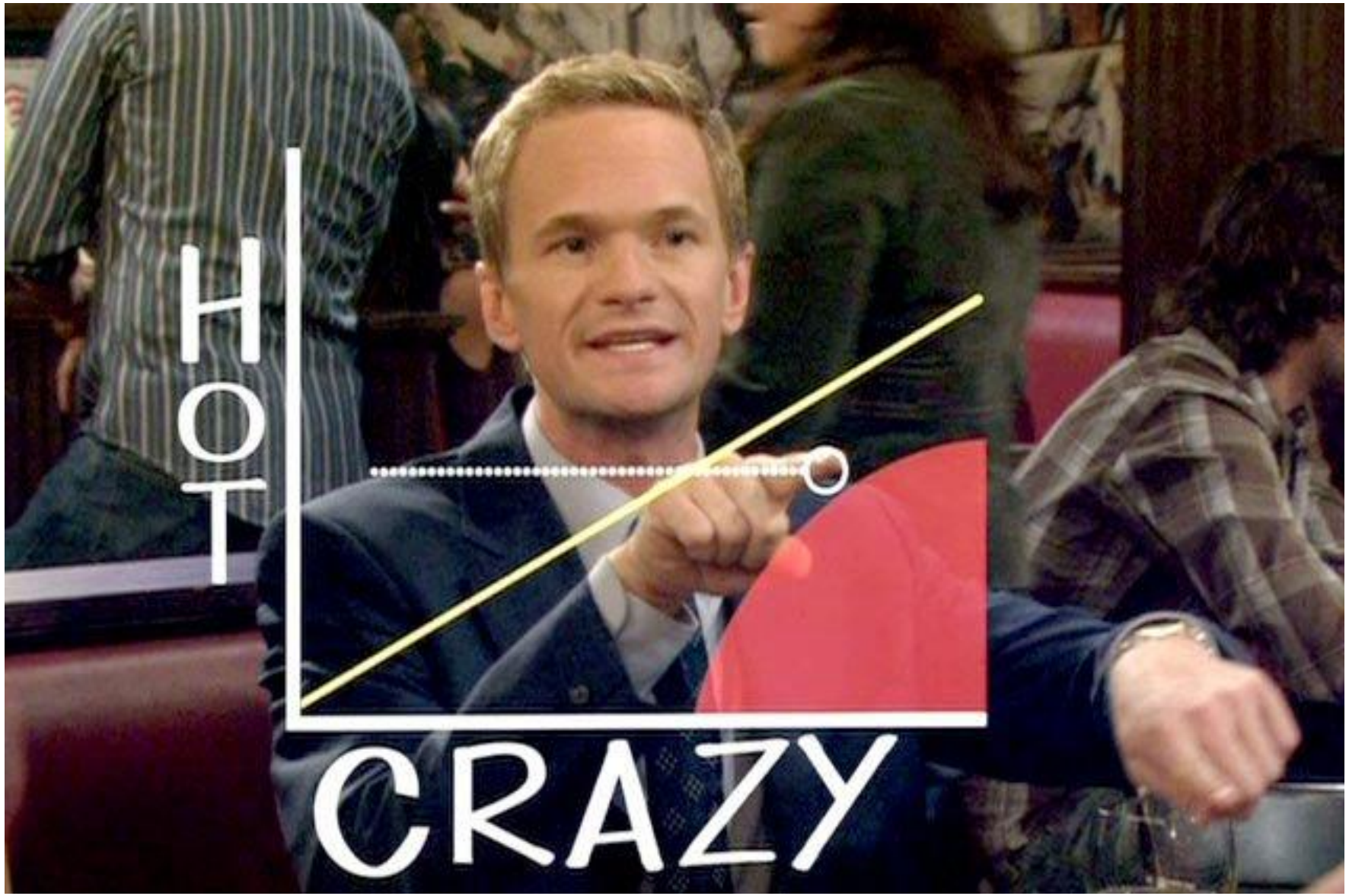
.Net Framework 4.7 & dotnet core
out var, pattern matching, Tuples,
Deconstruction, Wildcards, local functions, literal
improvements, ref returns, more inline
definitions, throw in expressions

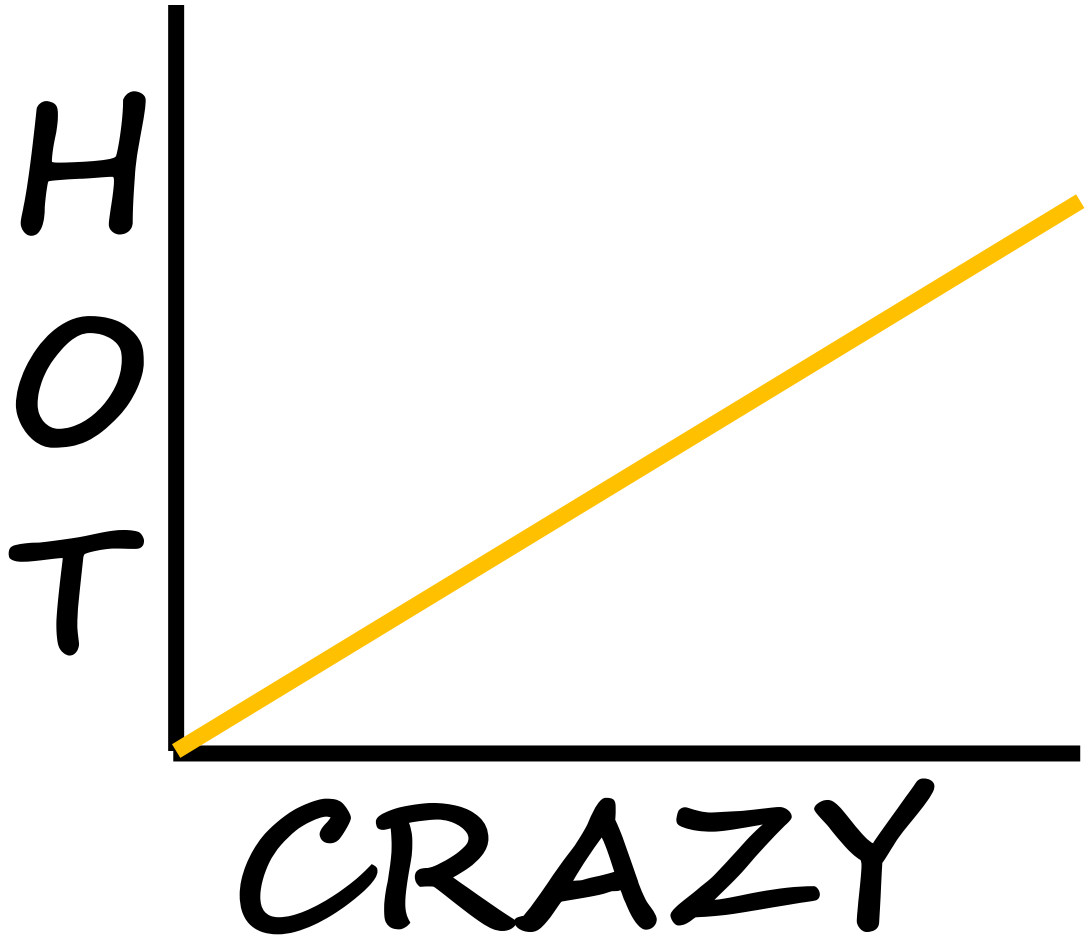
2019?

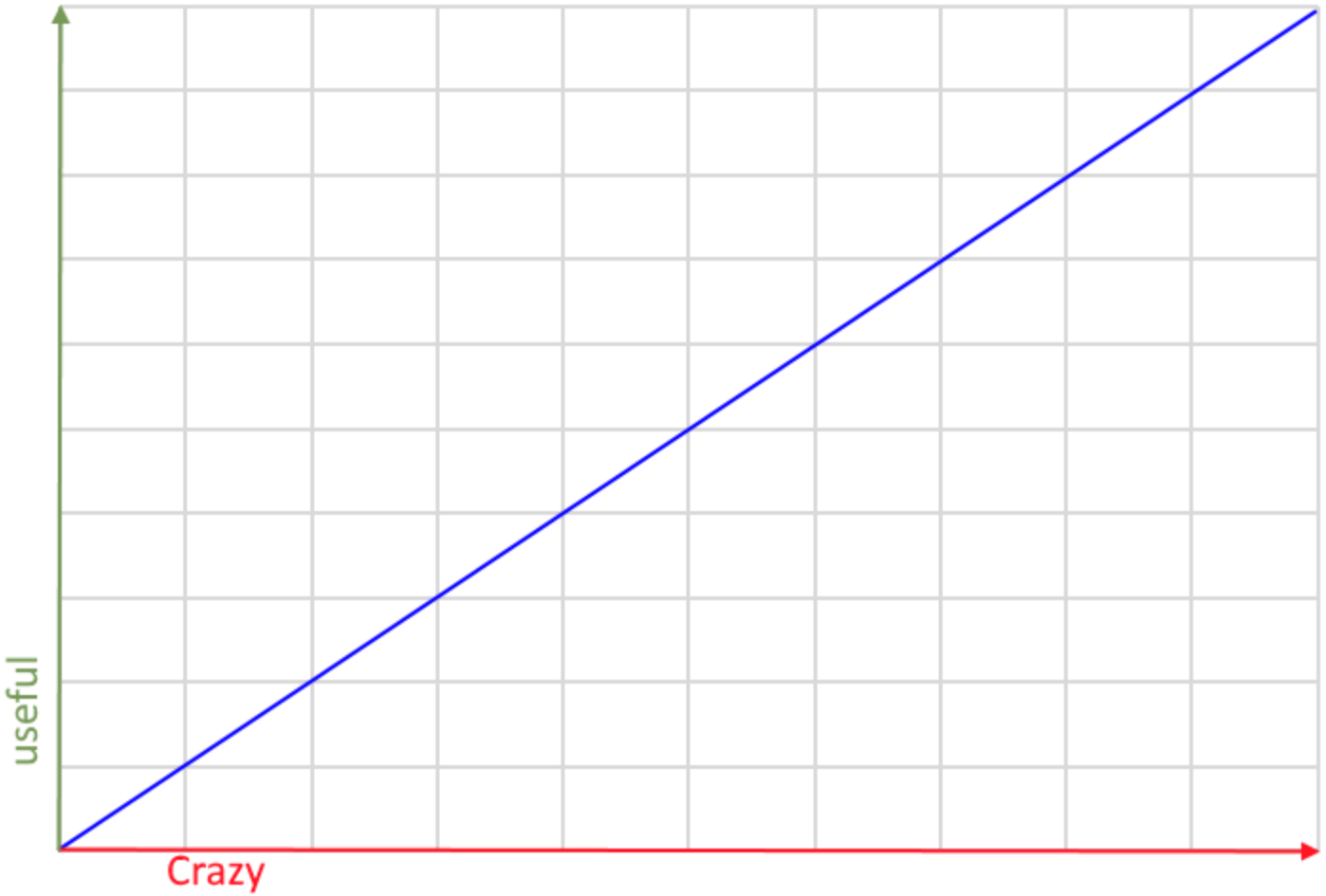
8.0

????




HOT
CRAZY


HOT
CRAZY


useful
Crazy

HELP US VOTING IN THE APP

http://bit.ly/netcoreapp

HOT CRAZY **C#**

Welcome to Hot Crazy C# APP

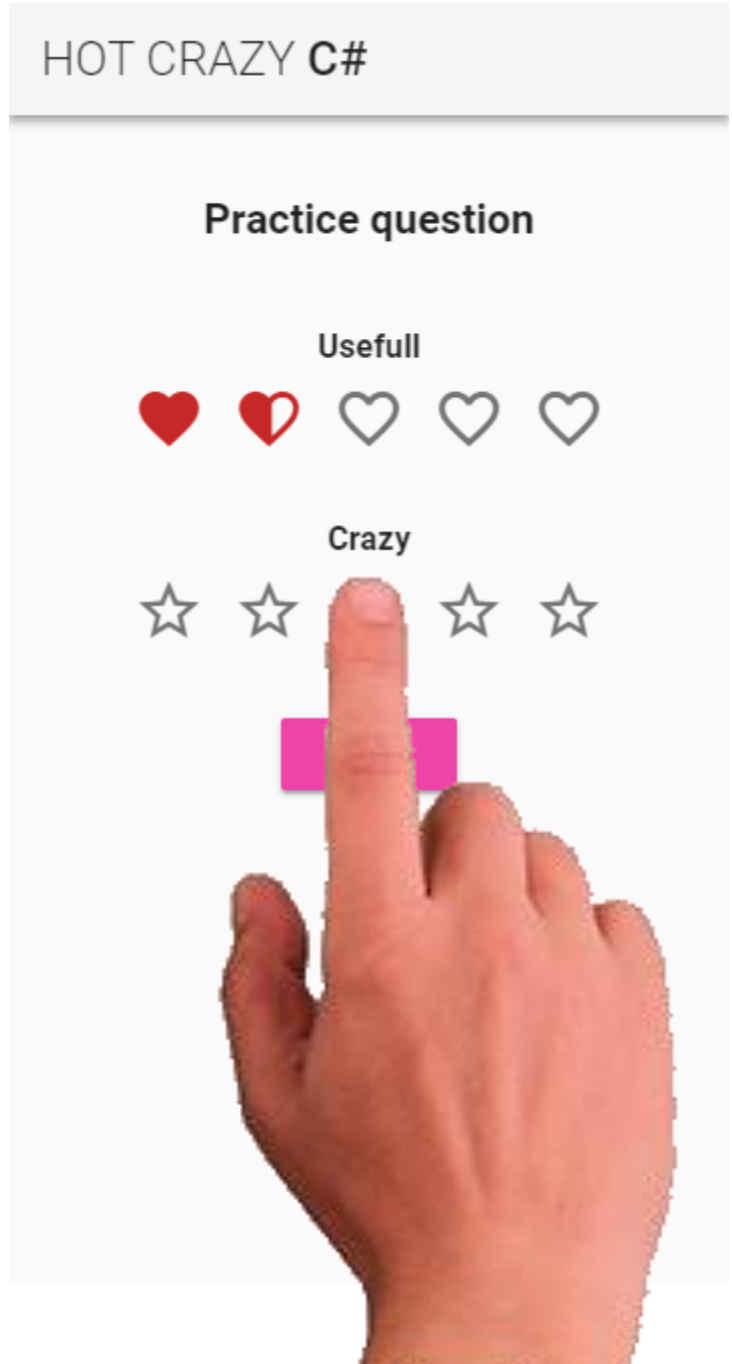
This application will help you voting the session poll. Please introduce one valid contact information (like a twitter account or your email) in order to start the poll:

Contact information

@fernandoescolar

16 / 256

START


HOT CRAZY C#
Practice question
Usefull
Crazy

VOTING IN THE APP

00:20

Nullable reference types

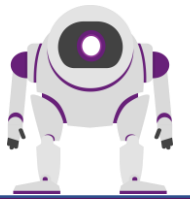
```
using static System.Console;
class Program
{
    static void Main(string[] args)
    {
        string s = null;
        WriteLine($"The first letter of {s} is {s[0]}");
    }
}
```

7.0

Exception Unhandled X

System.NullReferenceException: 'Object reference not set to an instance of an object.'

s was null.



Nullable reference types

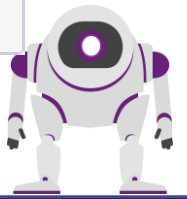
```
using static System.Console;
class Program
{
    static void Main(string[] args)
    {
        string? s = null;
        WriteLine($"The first letter of {s} is {s[0]}");
    }
}
```

7.0

class System.String

Represents text as a sequence of UTF-16 code units.

The type 'string' must be a non-nullable value type in order to use it as parameter 'T' in the generic type or method 'Nullable<T>'

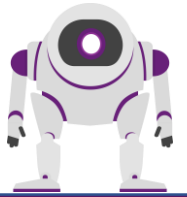


Nullable reference types

```
using static System.Console;
class Program
{
    static void Main(string[] args)
    {
        string s = null;
        WriteLine($"The first letter of {s} is {s[0]}");
    }
}
```

8.0

Warning!
Assignment of null to non-nullable reference type

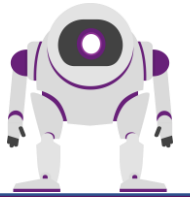


Nullable reference types

```
using static System.Console;
class Program
{
    static void Main(string[] args)
    {
        string? s = null;
        WriteLine($"The first letter of {s} is {s[0]}");
    }
}
```

8.0

Warning!
Possible null reference exception



Nullable reference types

```
using static System.Console;
class Program
{
    static void Main(string[] args)
    {
        string? s = null;
        WriteLine($"The first letter of {s} is {s[0] ?? 'null' }");
    }
}
```

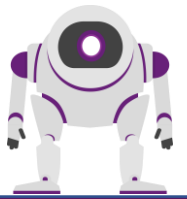
8.0

VOTING IN THE APP

00:20

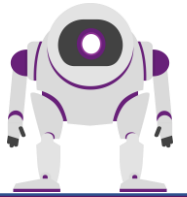
Async streams

```
async Task<int> GetOneResultAsync()
{
    var result = await GetFooAsync();
    if (result > 20) return result;
    else return -1;
}

async Task<IEnumerable<int>> GetManyResultsAsync()
{
    var list = new List<int>();
    int i = -1;
    do {
        list.Add(await GetOneAsync());
    } while (i > 0);

    return list;
}
```



Async streams

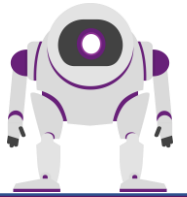
```
async Task<int> GetOneResultAsync()
{
    var result = await GetFooAsync();
    if (result > 20) return result;
    else return -1;
}

async Task<IEnumerable<int>> GetManyResultsAsync()
{
    int i = -1;
    do
    {
        i = await GetOneAsync());
        yield return i;
    } while (i > 0);
}
```



Async streams

```
async Task<int> GetOneResultAsync()
{
    var result = await GetFooAsync();
    if (result > 20) return result;
    else return -1;
}
```

```
async IAsyncEnumerable<int> GetManyResultsAsync()
{
    await foreach (var result in GetFooAsync())
    {
        if (result > 20) yield return result;
    }
}
```

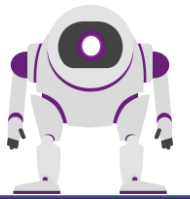


VOTING IN THE APP

00:20

Rangos e índices

```
Person[] people = {
    new ("Elena"), new ("Armando"), new ("Dolores" ), new ("Aitor"),
    new ("Leia"),  new ("Vader"),   new ("Yoda"),     new ("Skywalker"),
};

foreach (var p in people[0..3]) Console.Write($"{p.FirstName} ,");
// Elena, Armando, Dolores, Aitor

Index indexStart = 1;
Index indexEnd = ^5;
Range range = people[1..^5] ←→ people[indexStart..indexEnd])

// Leia, Vader, Yoda, Skywalker

foreach (var p in people[6..]) Console.Write($"{p.FirstName} ,");
// Yoda, Skywalker

foreach (var p in people[..]) Console.Write($"{p. FirstName} ,");
// all
```

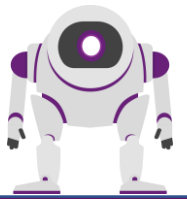


VOTING IN THE APP

00:20

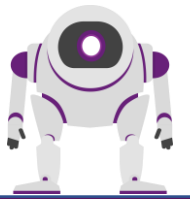
Patrones recursivos

```
IEnumerable<string> GetEnrollees()
{

    foreach (var p in People)
    {
        var student = p as Student;
        if (student != null && !student.Graduated)
        {
            var name = student.Name;
            yield return name;
        }
    }
}
```



Patrones recursivos

```
IEnumerable<string> GetEnrollees()
{

    foreach (var p in People)
    {
        if (p is Student { Graduated: false, Name: string name }) yield return name;
    }
}
```



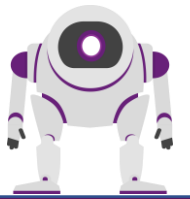
VOTING IN THE APP

00:20

Expresiones switch

```
object o = new Point(X: "10", Y: "5");

switch (o)
{
    case Point p when p.X == 0 && p.Y == 0: return "origin";
    case Point p:                           return $"{p.X}, {p.Y}";
    default:                                return "unknown";
};
```

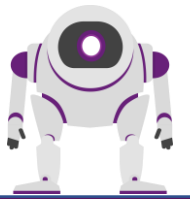
7.0



Expresiones switch

```
object o = new Point(X: "10", Y: "5");

return o switch
{
    Point p when p.X == 0 && p.Y == 0  => "origin",
    Point p                            => ${p.X}, {p.Y}",
    _                                  => "unknown"
};
```

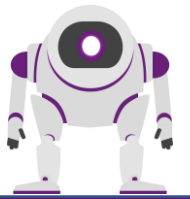
8.0



Expresiones switch

```
object o = new Point(X: "10", Y: "5");

return o switch
{
    Point { X: 0, Y: 0 }          => "origin",
    Point { X: var x, Y: var y } => $"{x}, {y}",
    _                             => "unknown"
};
```

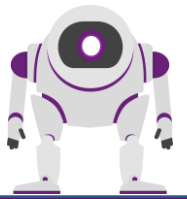
8.0



Expresiones switch

```
object o = new Point(X: "10", Y: "5");

return o switch
{
    Point(0, 0)          => "origin",
    Point(var x, var y) => $"{x}, {y}",
    _                    => "unknown"
};
```

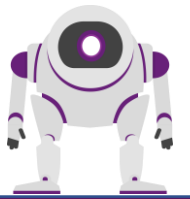
8.0



Expresiones switch

```
var p = new Person(firstName: "Elena", middleName: "Nito", lastName: "del Bosque");

return (p.FirstName, p.MiddleName, p.LastName) switch
{
    (string f, null      , null     )  => $"{f}"                 // Elena
    (string f, null      , string  l)  => $"{f} {l}",            // Elena del Bosque
    (string f, string    m, null     )  => $"{f} {m}",            // Elena Nito
    (string f, string    m, string  l)  => $"{f} {m[0]}. {l}",    // Elena N. del Boque
    _                                   => $"Sr./Sra.",          // Sr./Sra.
};
```

8.0



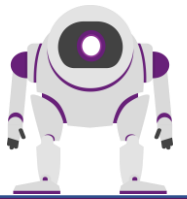
VOTING IN THE APP

00:20

Constructores implícitos

```
Person[] people =
{
    new Person("Elena", "Nito", "del Bosque"),
    new Person("Armando", "Bronca", "Segura"),
    new Person("Dolores", "Cabeza", "Baja"),
    new Person("Aitor", "Tilla", "del Bosque"),
};
```

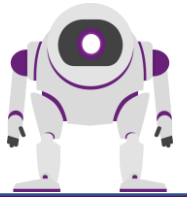
7.0



Constructores implícitos

```
Person[] people =
{
    new ("Elena", "Nito", "del Bosque"),
    new ("Armando", "Bronca", "Segura"),
    new ("Dolores", "Cabeza", "Baja"),
    new ("Aitor", "Tilla", "del Bosque"),
};
```

8.0



VOTING IN THE APP

00:20

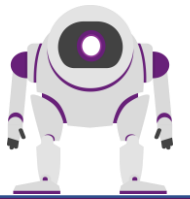
Declaración de “using”

```
static void Main(string[] args)
{
    using (var disposable = CreateDisposable(args))
    {
        ...
    } // disposable is disposed here
}
```

7.0

```
static void Main(string[] args)
{
    using var disposable = CreateDisposable(args);
    ...
} // disposable is disposed here
```

8.0



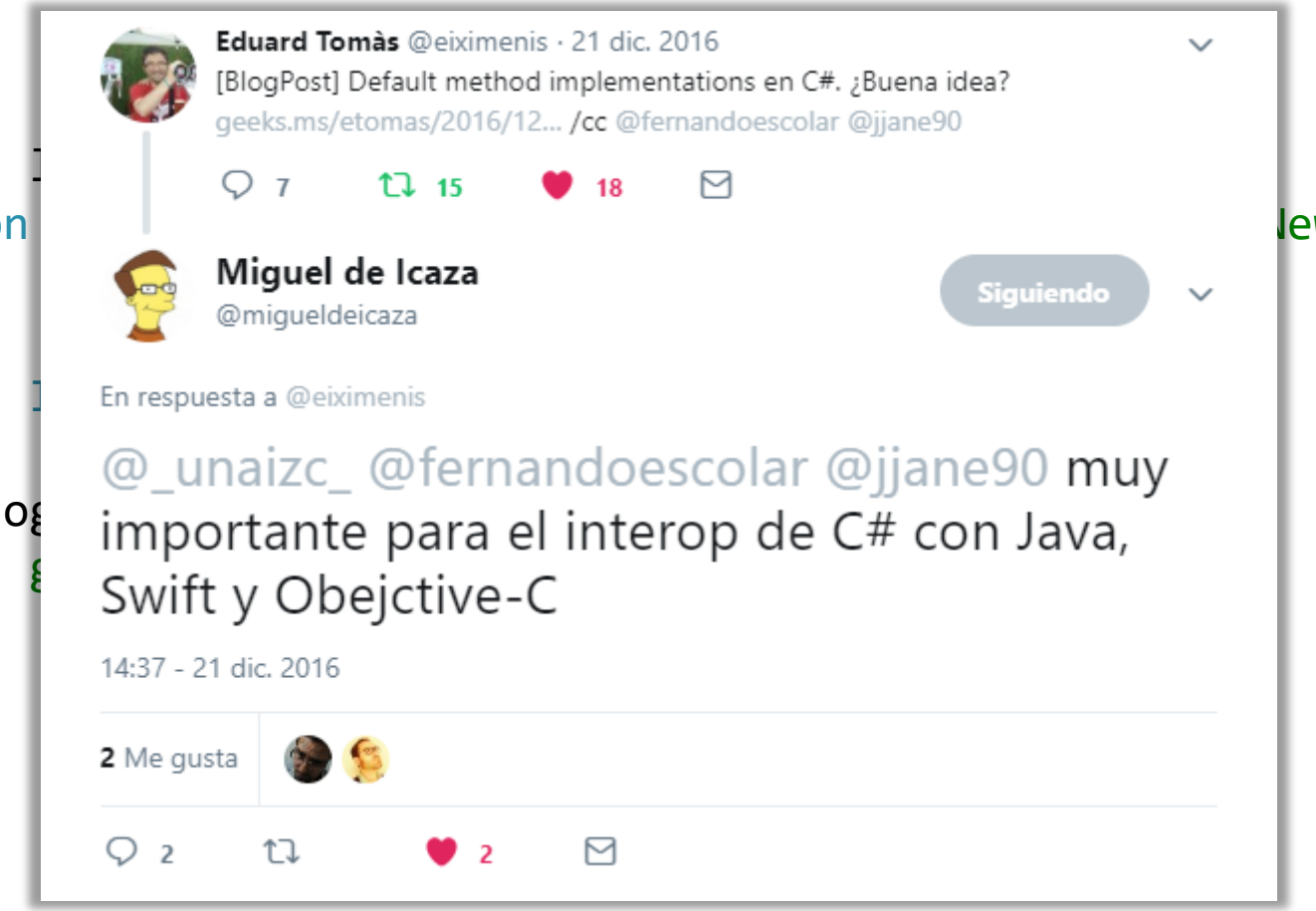
VOTING IN THE APP

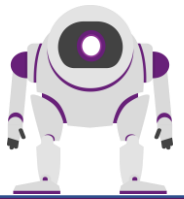
00:20

Interfaces con implementación

```
interface ILogger
{
    void Log(LogLevel l
    void Log(Exception
}

class ConsoleLogger : I
{
    public void Log(Log
    // Log(Exception) {
}
```

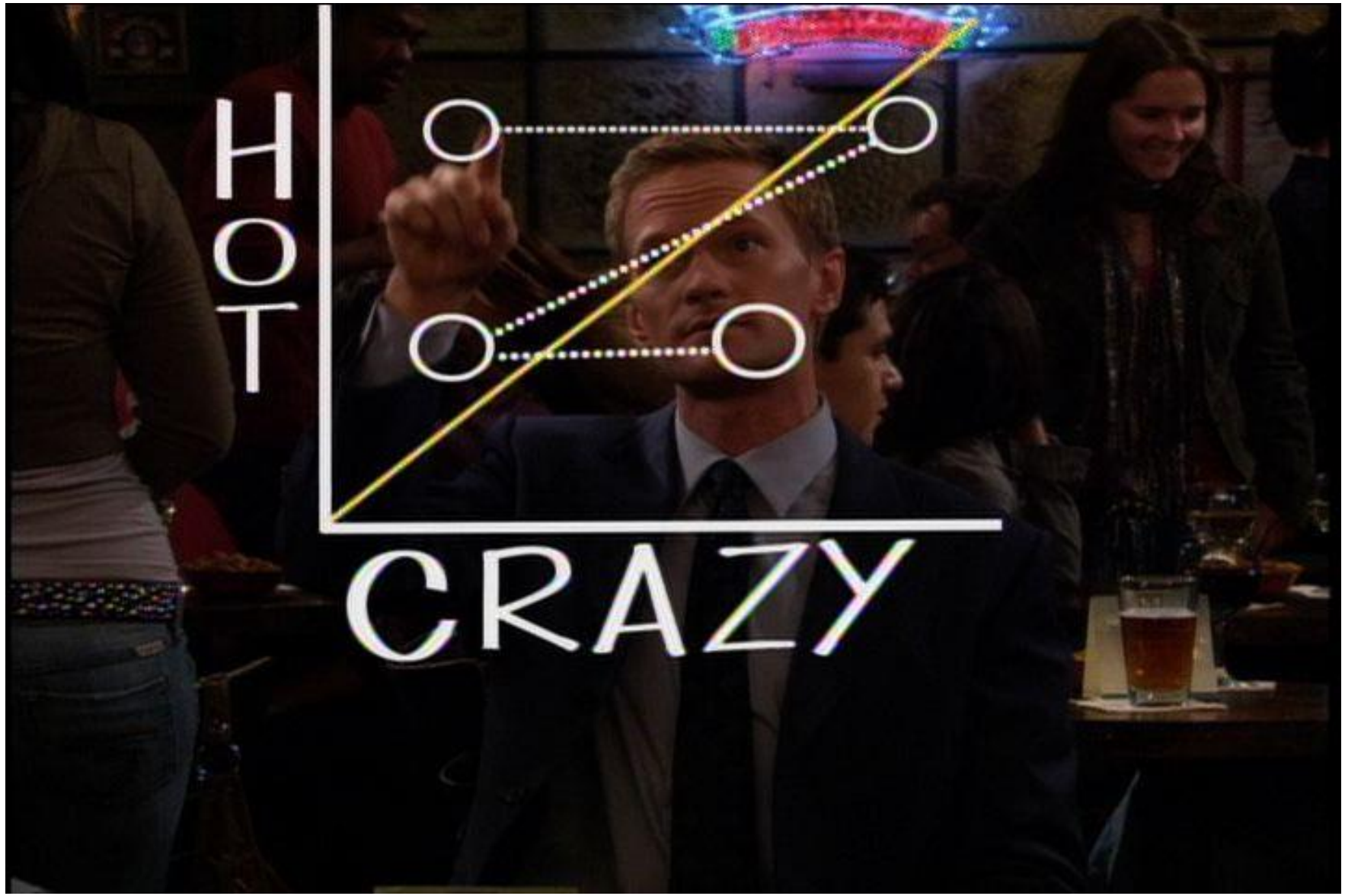
```
ew overload
```





VOTING IN THE APP

00:20


HOT
CRAZY

Sponsors

Microsoft

avanade

plain concepts

iTEQUIA

encamina
PIENSA EN COLORES

東京 TOKIOTA

devs dna

3ASIDE Consultors

intelequia
SOFTWARE SOLUTIONS

minsait
An Indra company

Insight

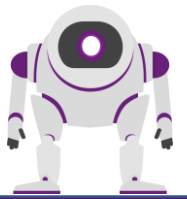
everis
an NTT DATA Company

Sothis

IRON HACK
BARCELONA

GIN MG
CASA FUNDADA
EN 1835

steema
software



Más información:

info@netcoreconf.com
@Netcoreconf

Visítanos en:
2019.Netcoreconf.com