.NET Conference 2016 Spain

Microsoft

#dotNetSpain2016

50 piezas de Lego

No recomendada para mayores de 18 años

Isabel Cabezas
@isabelcabezasm
Beatriz García
@BeRoces

Microsoft

#dotNetSpain2016

Patrocinadores

Colaboradores

#dotNetSpain2016

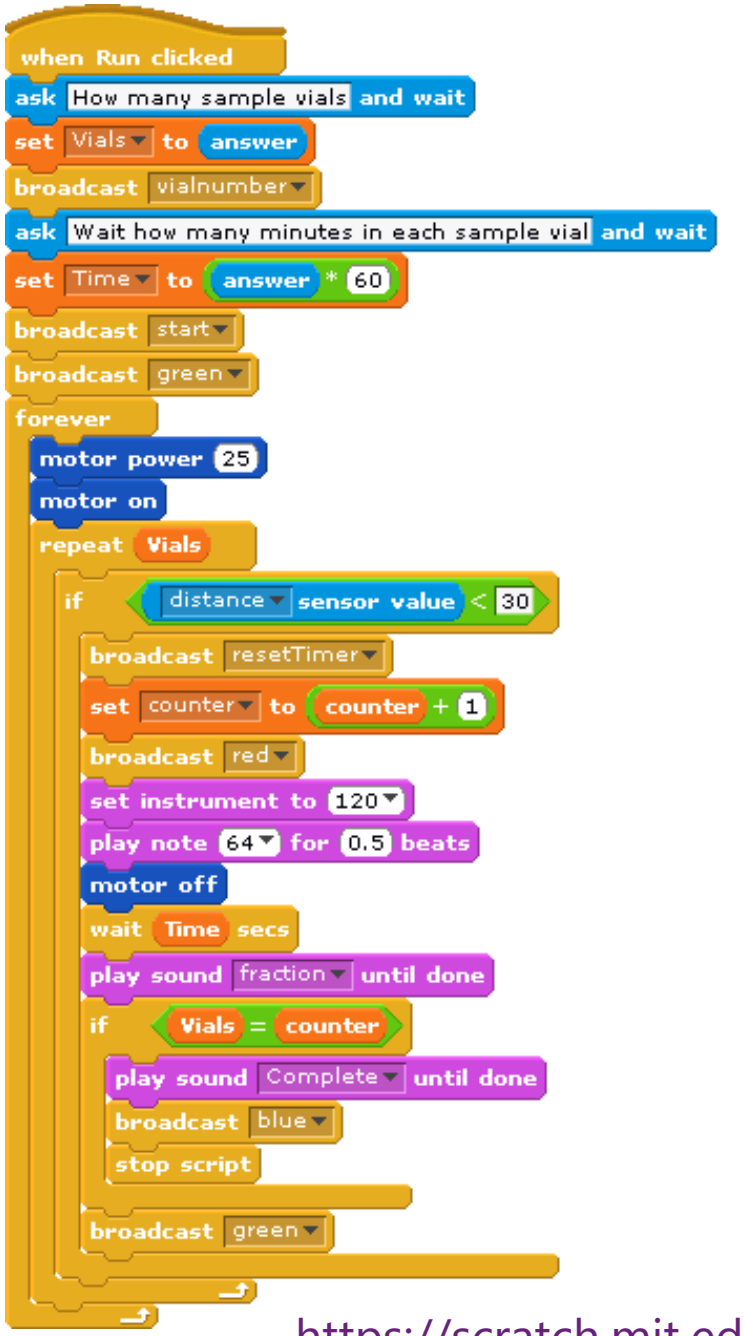
Agenda

- Lego Mindstorms
- ¿Es solo un juego de niños?
- ¡Todos queremos jugar!
- DEMO
- Otros detalles técnicos

¿Quién no ha querido un Lego de pequeño?

Lego WeDo

Para niños
de 7 a 11 años

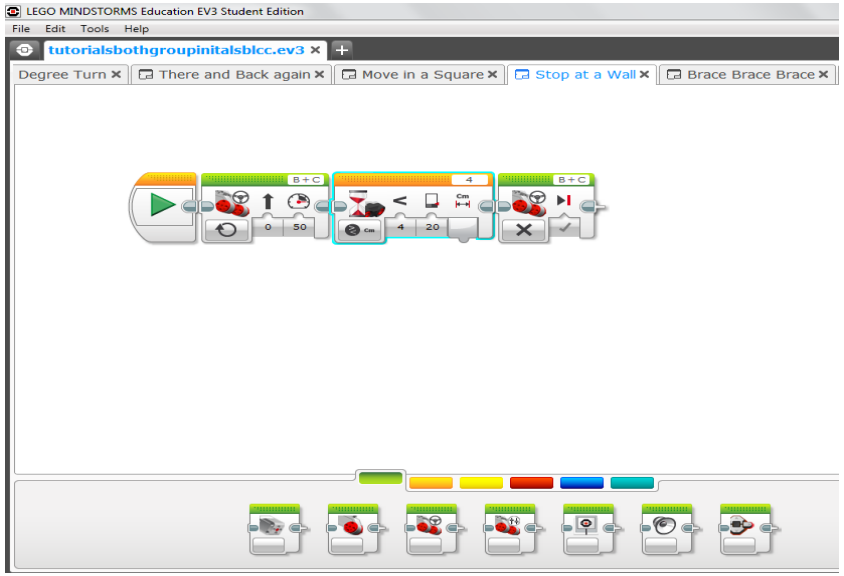


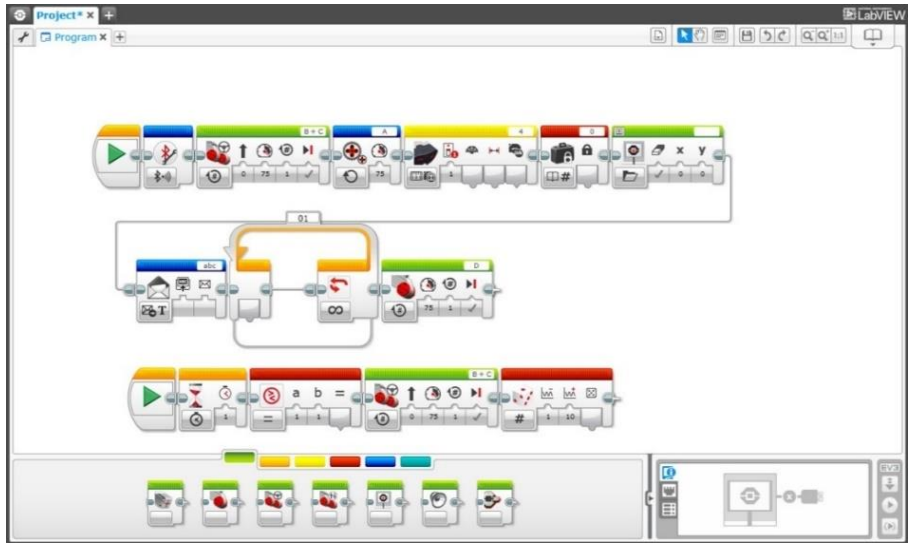
https://scratch.mit.edu/

#dotNetSpain2016

Lego MindStorms

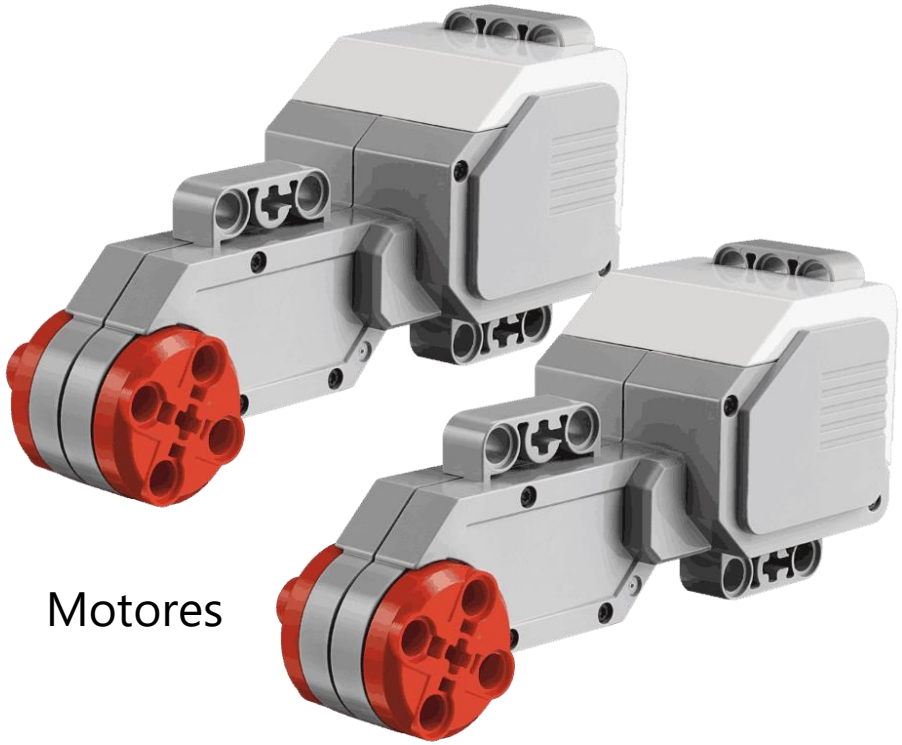
A partir de 10 años





#dotNetSpain2016

Lego MindStorms

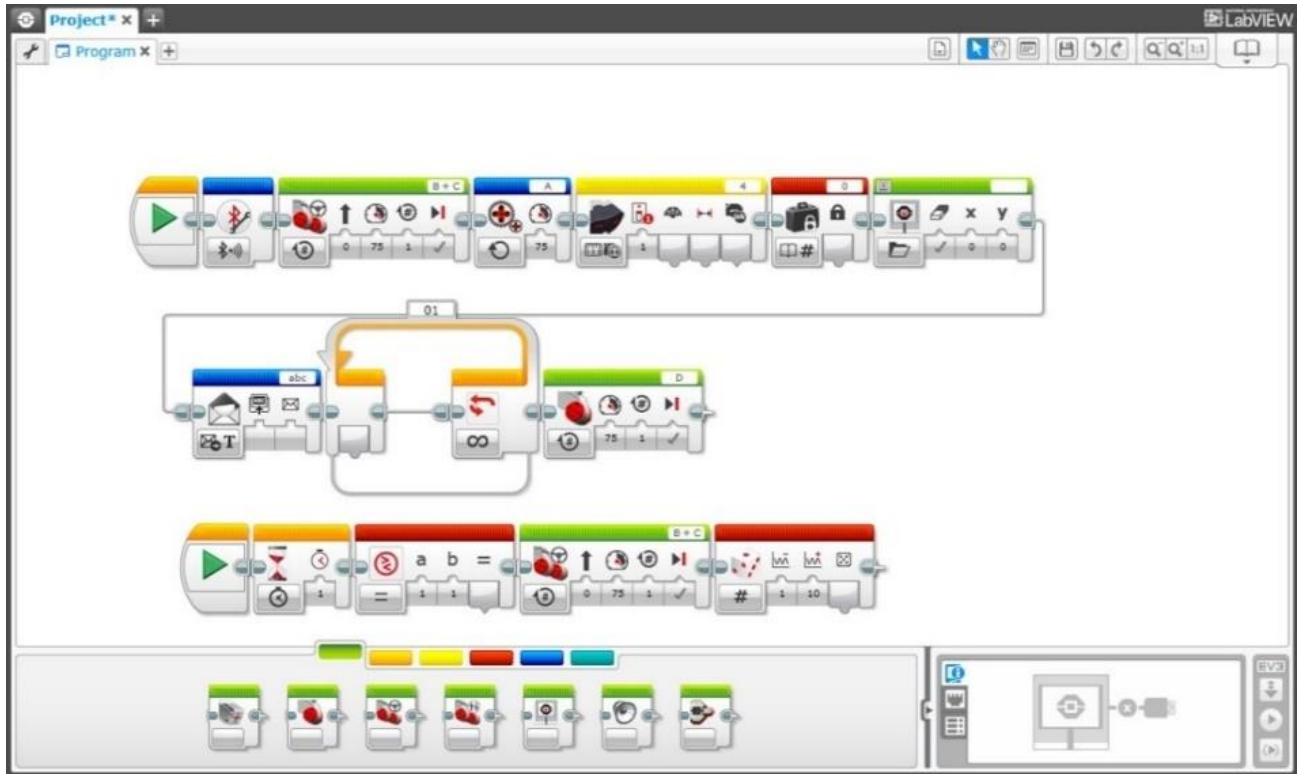


Motores

IR

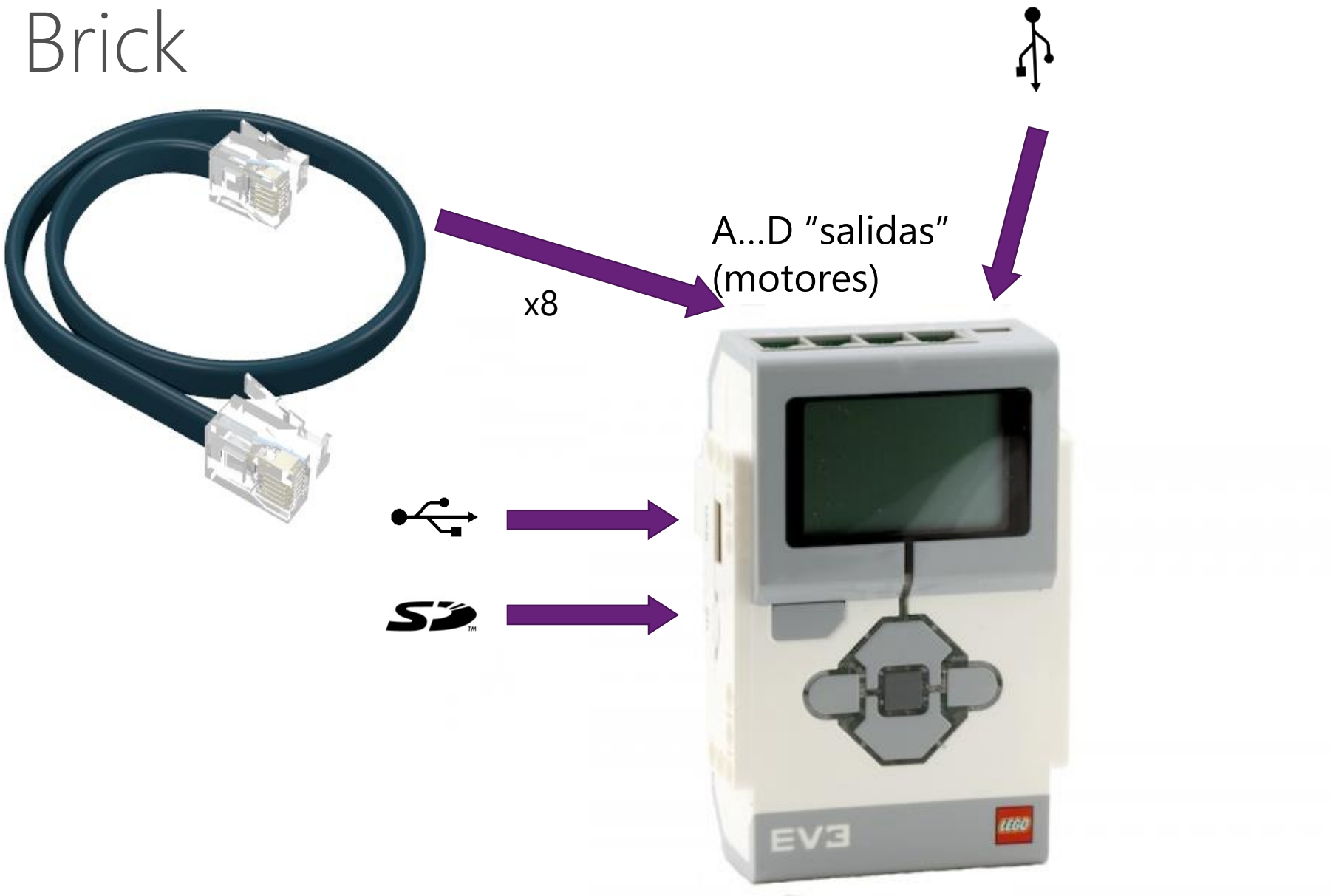
Brick

#dotNetSpain2016

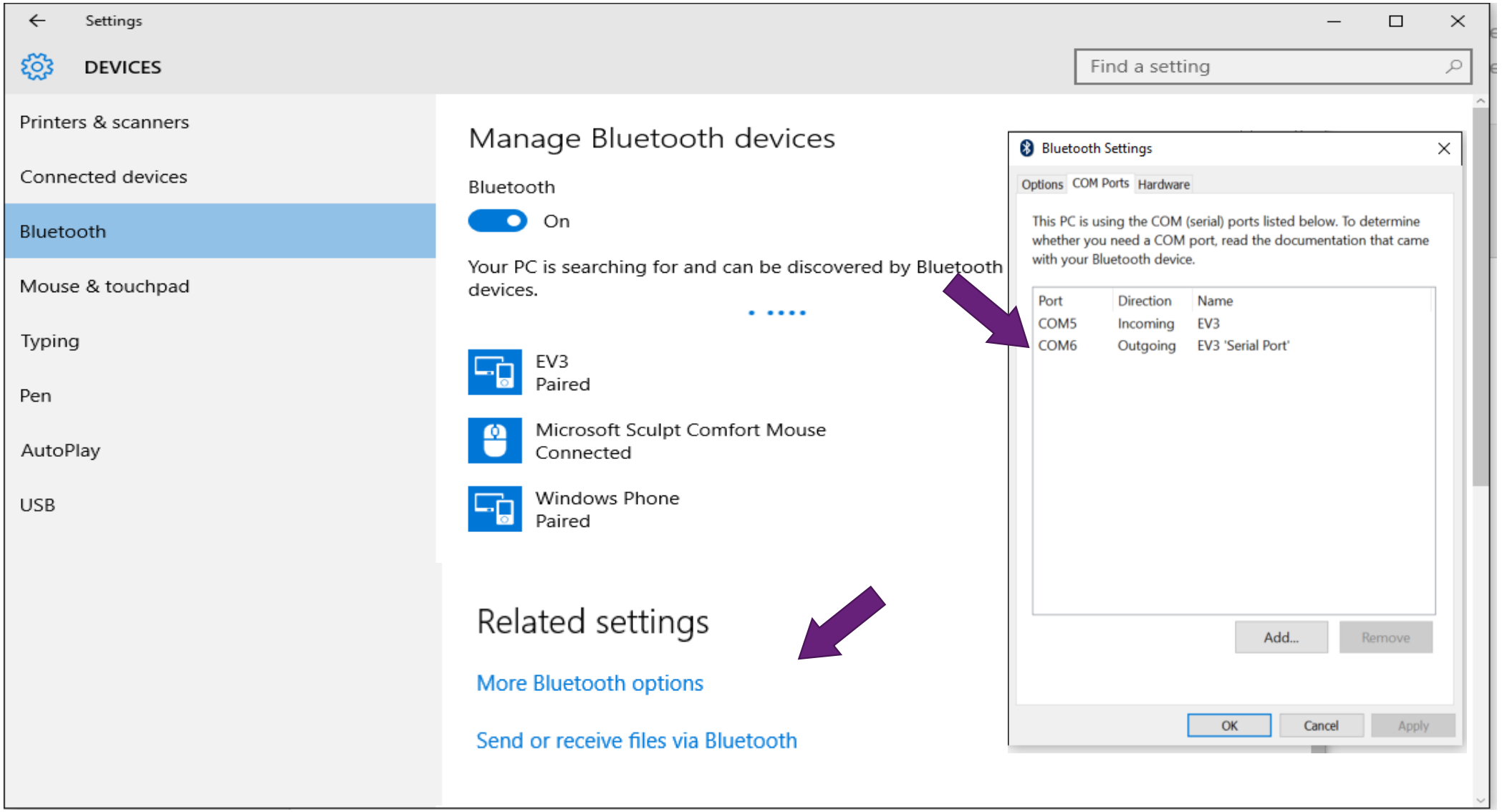


#dotNetSpain2016

El Brick

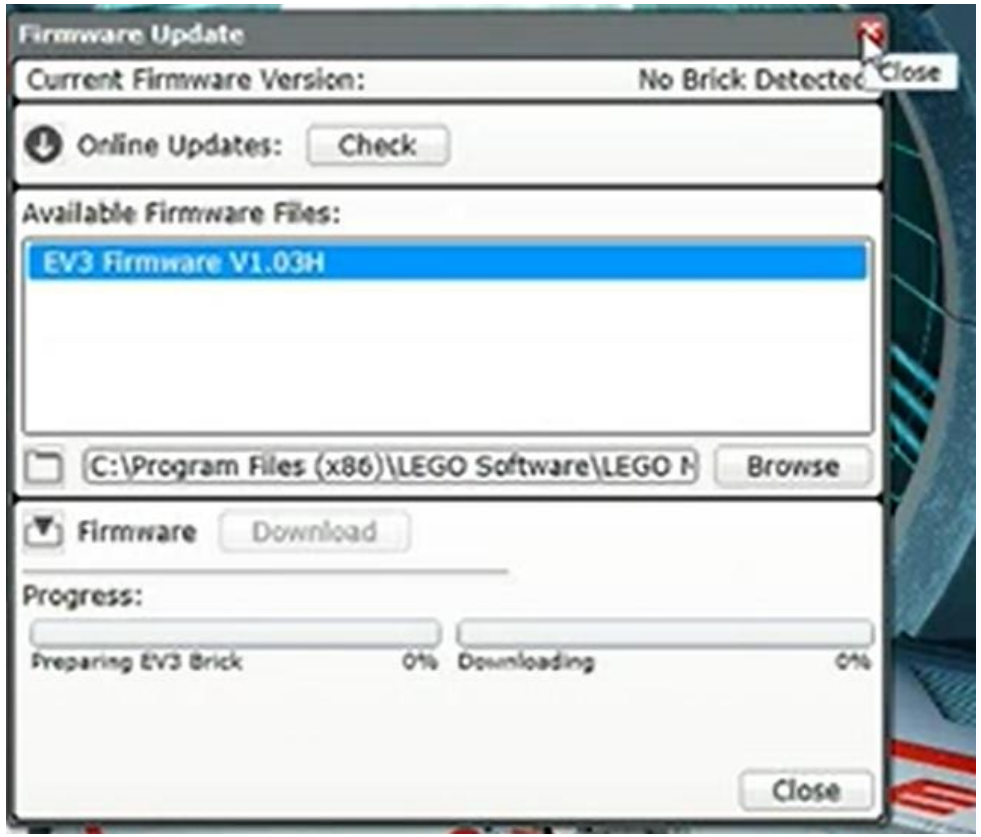


Conectar por BT desde Windows 10



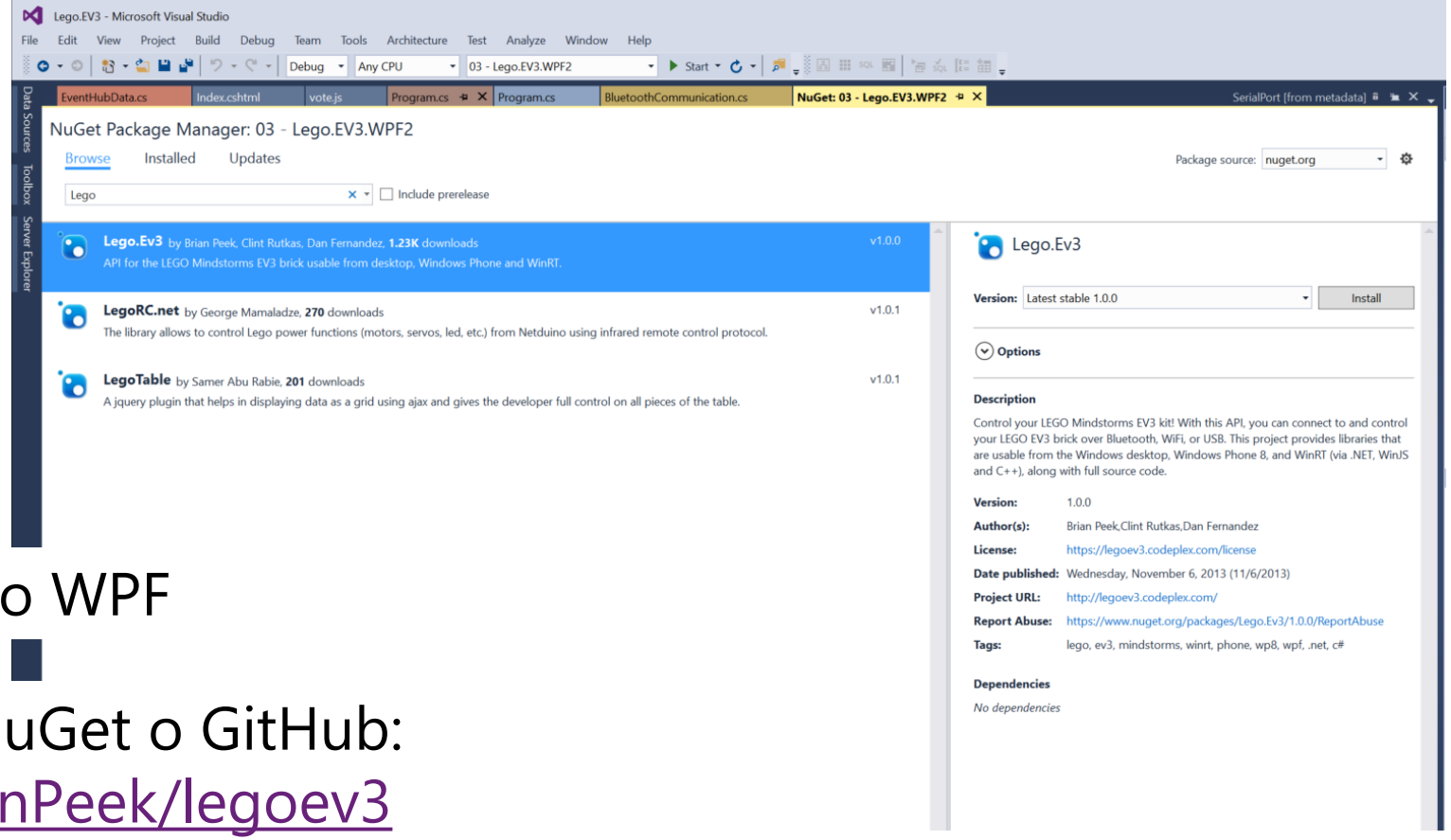
#dotNetSpain2016

No olvides actualizar el firmware



#dotNetSpain2016

Para empezar...

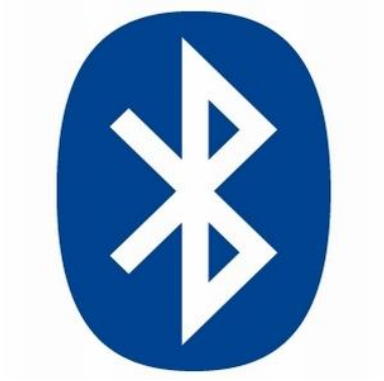


- Crear un proyecto nuevo WPF
- Descargar SDK desde NuGet o GitHub: https://github.com/BrianPeek/legoev3
- Using it!

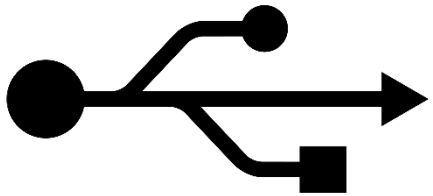
```
using System.Windows.Shapes;
using Lego.Ev3.Core;
using Lego.Ev3.Desktop;
```

#dotNetSpain2016

Conectar con nuestro robot



```
System.IO.Ports
SerialPort : Component

public BluetoothCommunication(string port)
{
        _serialPort = new SerialPort(port, 115200);
}
```



connect()
disconnect()
SerialPortDataReceived()
WriteAsync()

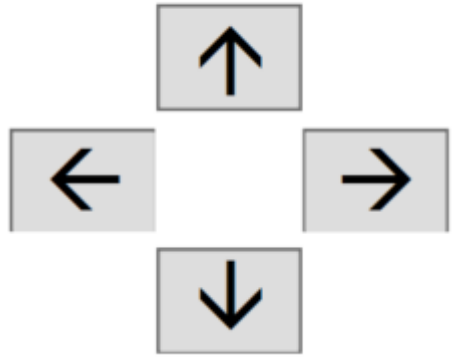
(Accesorio no incluido)

#dotNetSpain2016

Hello Piiiiiiiiiiiiiiiiiii!

```
Brick _brick;

_brick = new Brick(new BluetoothCommunication("COM6"));
_brick.BrickChanged += _brick_BrickChanged;
await _brick.ConnectAsync();
await _brick.DirectCommand.PlayToneAsync(100, 3, 300);
```

Motores

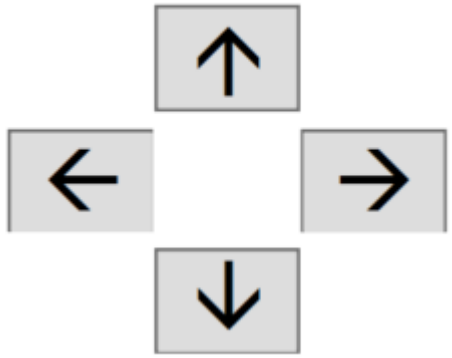


```
private void ForwardButton_Click(object sender, RoutedEventArgs e){
      _brick.DirectCommand.TurnMo
}
```

TurnMotorAtPowerAsync
TurnMotorAtPowerForTimeAsync
TurnMotorAtSpeedAsync
TurnMotorAtSpeedForTimeAsync

(awaitable) Task DirectCommand.TurnMotorAtPowerAsync(OutputPort ports, int power)
Turn the motor connected to the specified port or ports at the specified power.
Usage:
await TurnMotorAtPowerAsync(...);

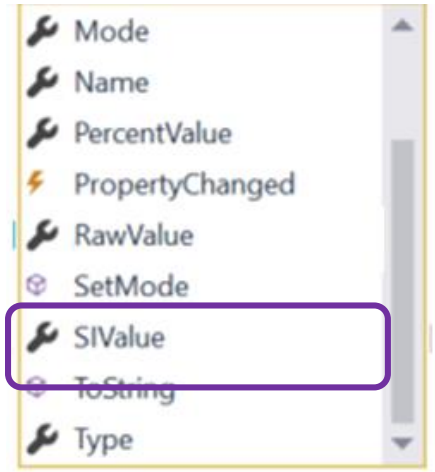
Motores



```
private void ForwardButton_Click(object sender, RoutedEventArgs e) {

_brick.DirectCommand.TurnMotorAtPowerForTimeAsync(
                    OutputPort.B | OutputPort.C, _forward, _time, false);

}
```

Hello Sensors!

```
private void brick_BrickChanged(object sender, BrickChangedEventArgs e) {

      txtDistance.Content = e.Ports[InputPort.Four].

}
```



Play Sounds

```
//upload file
await _brick.SystemCommand.CopyFileAsync("Overpower.rsf",
                                         "../prjs/Tester/Overpower.rsf");


//play file
await _brick.DirectCommand.PlaySoundAsync(50,
                                  "../prjs/Tester/Overpower");
```

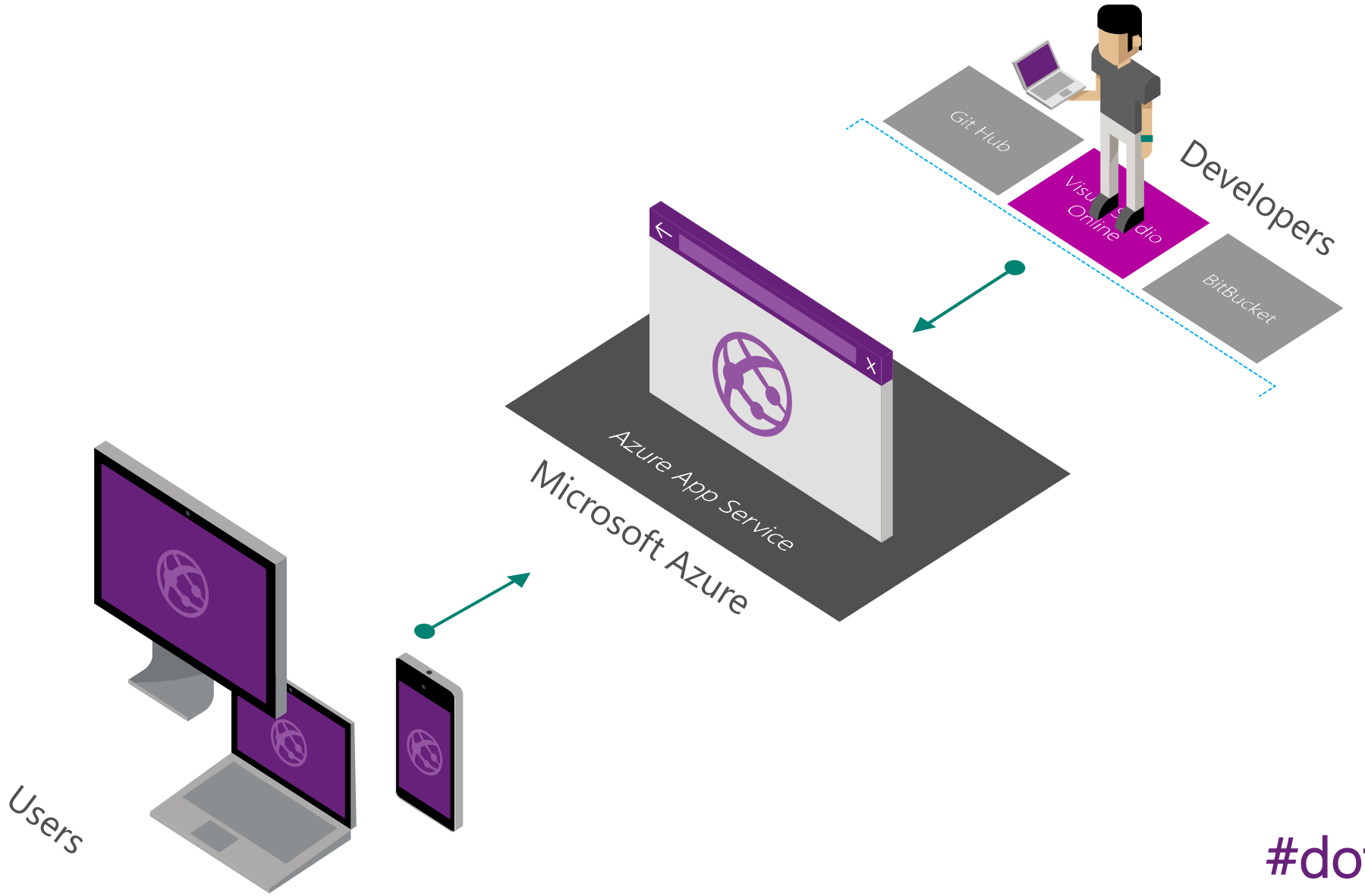
DEMO

Web App

Movimiento

#dotNetSpain2016

Azure Web Apps



#dotNetSpain2016

Votaciones individuales

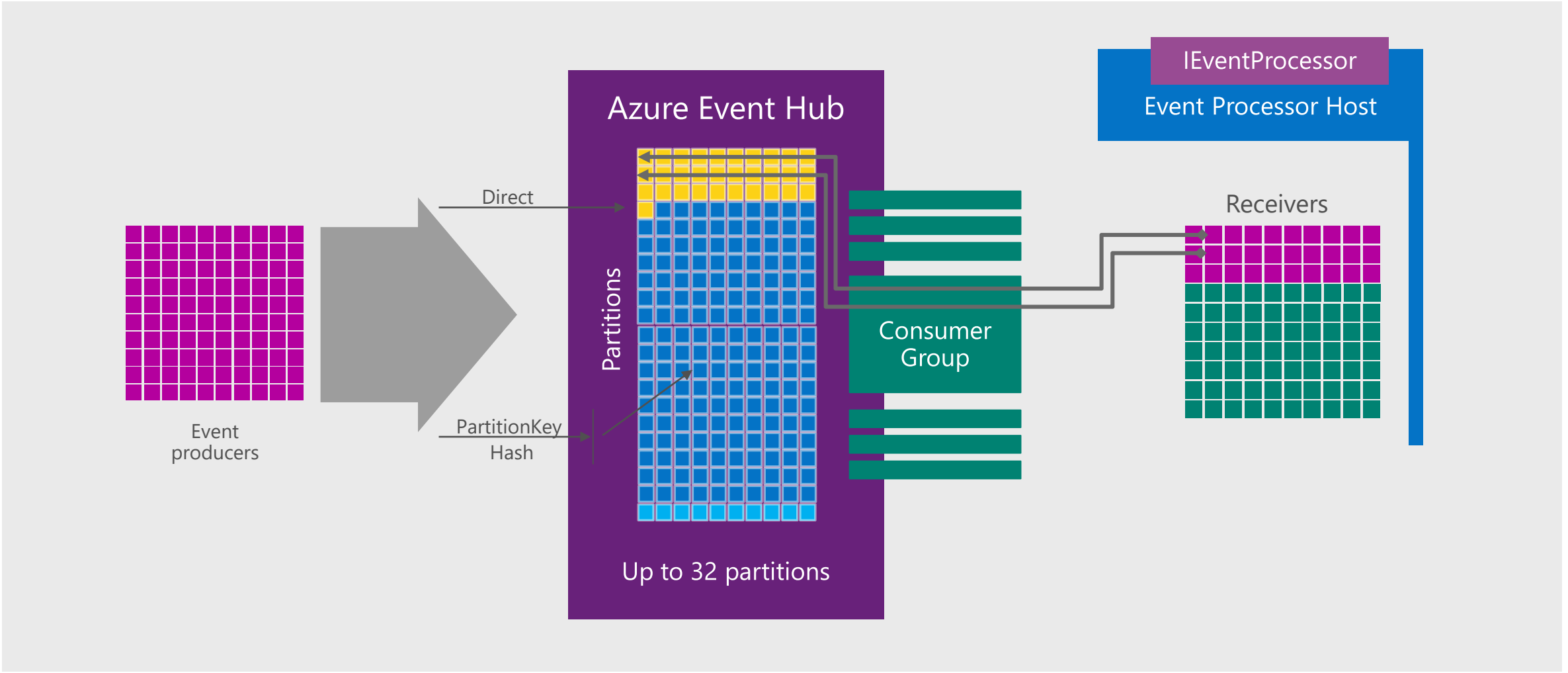
Event Hub

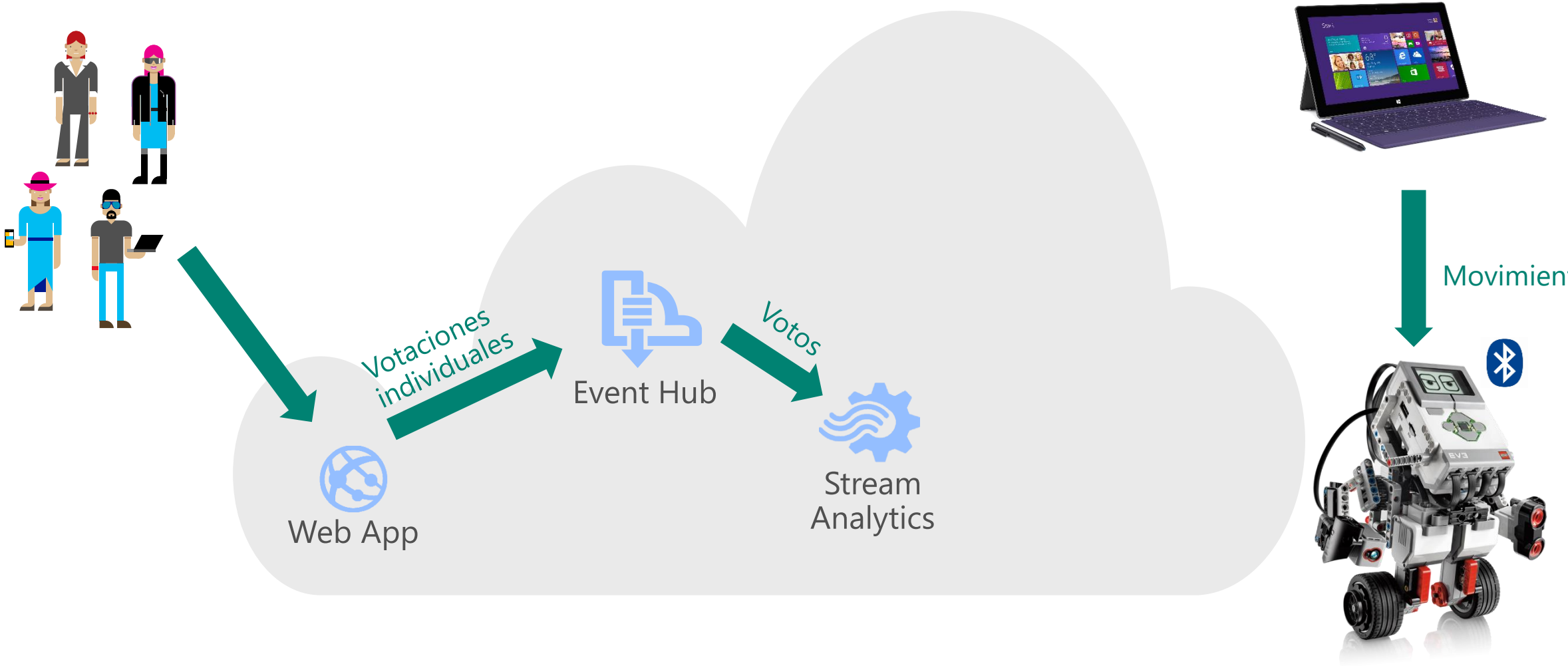
Web App

Movimiento

#dotNetSpain2016

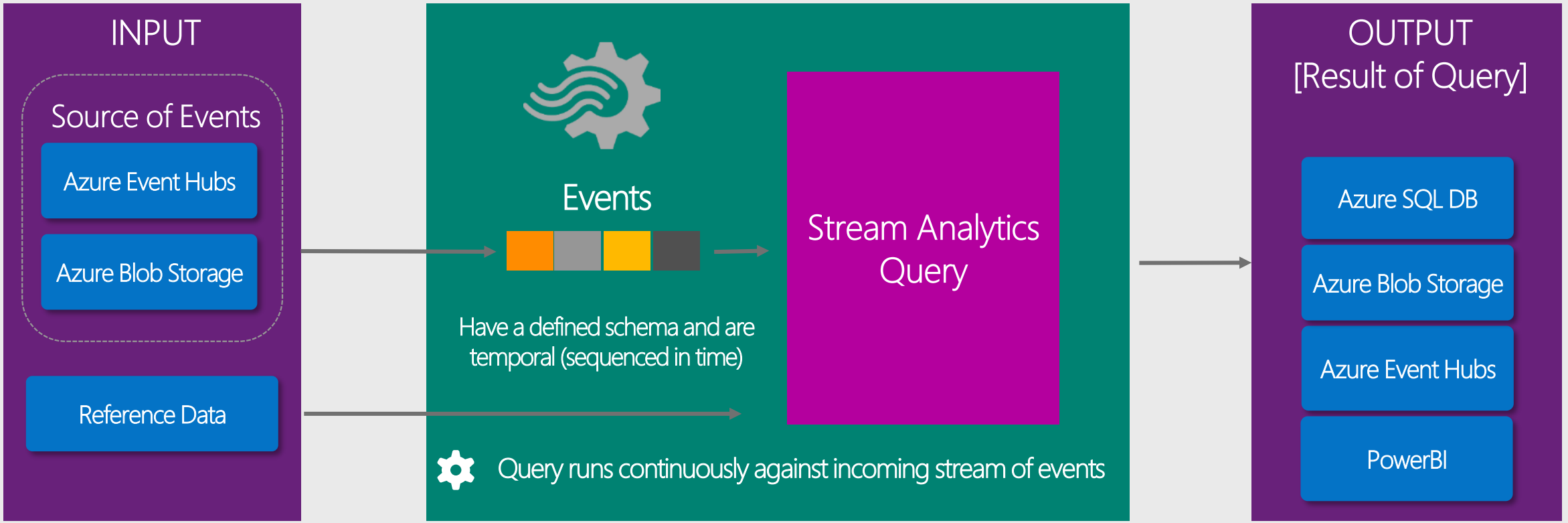
Service Bus Event Hub





#dotNetSpain2016

Stream Analytics



Votaciones individuales

Web App

Event Hub

Votos

Stream Analytics

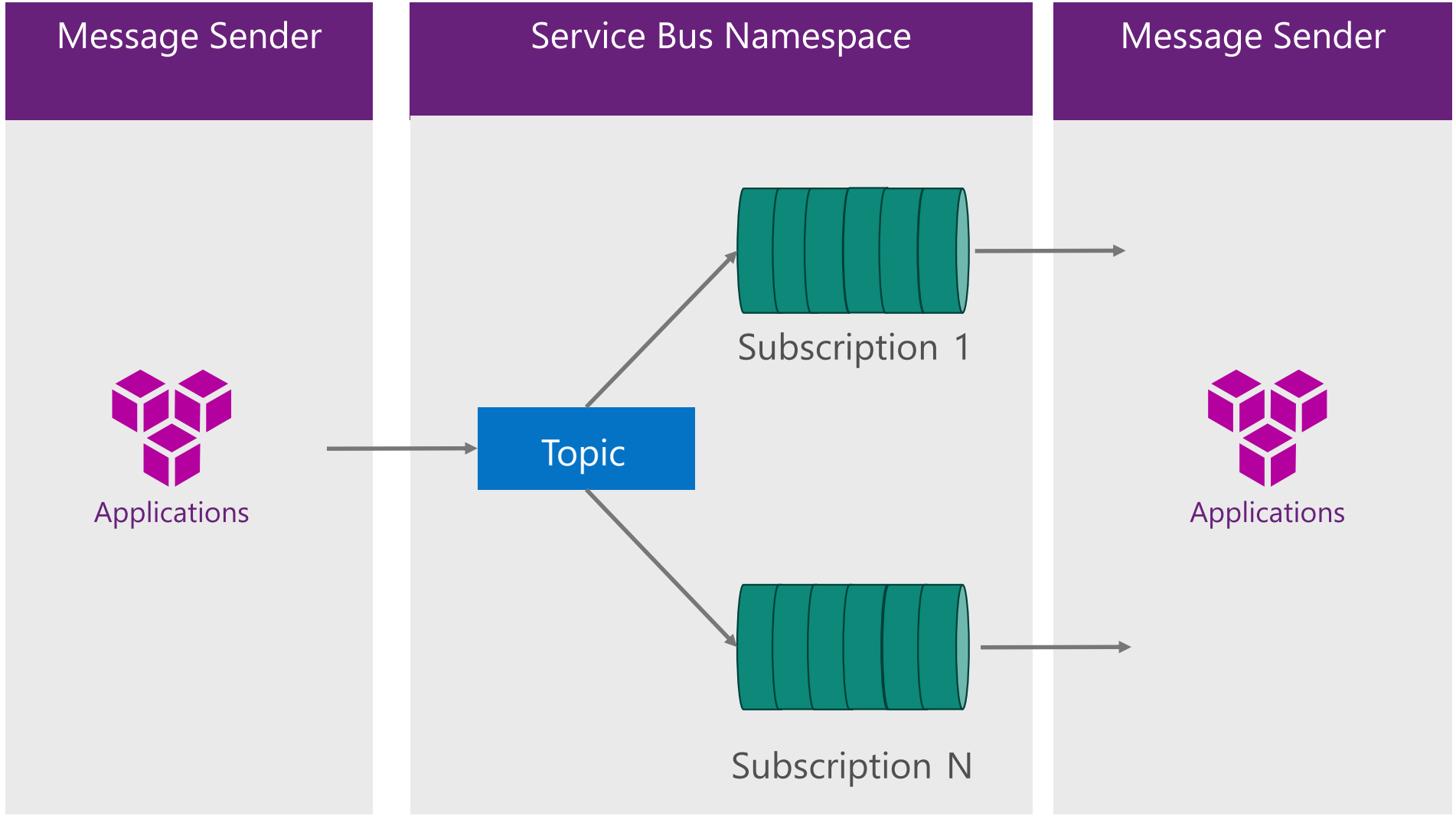
Resultado votación

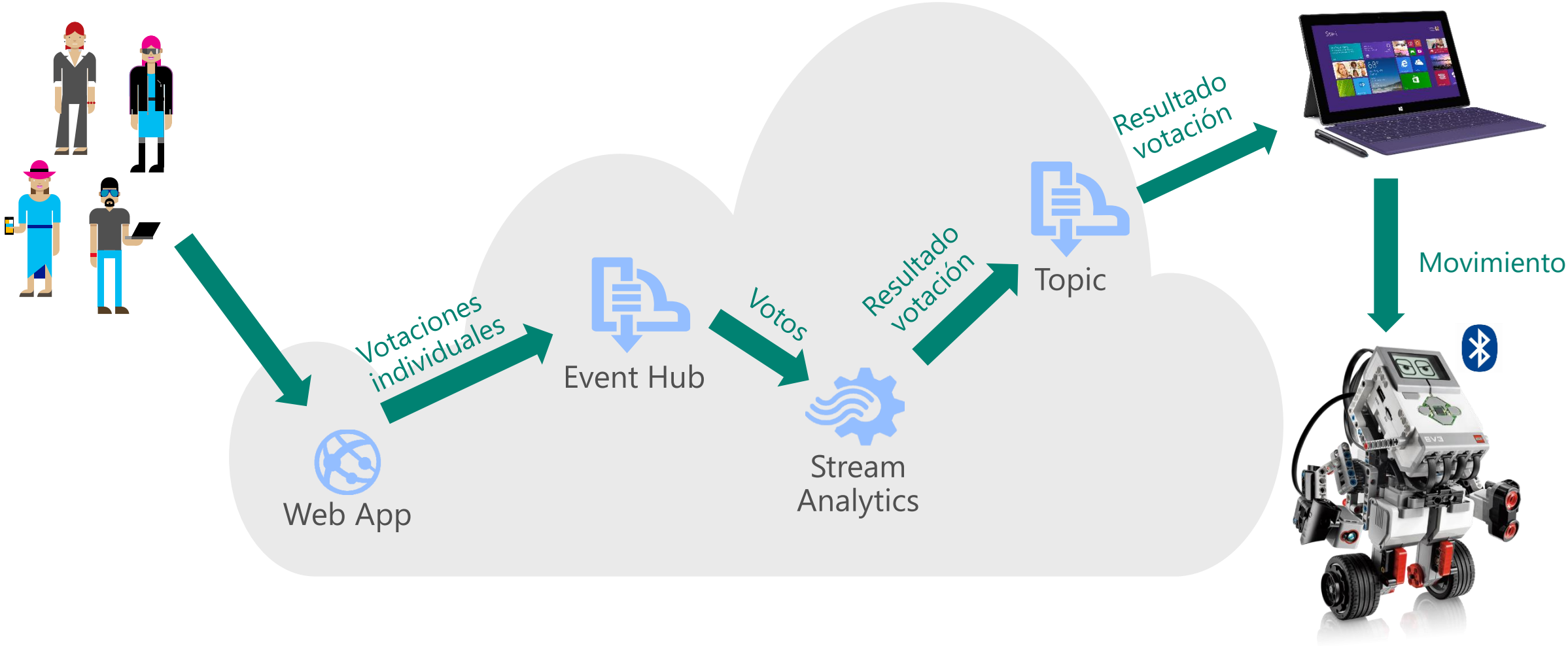
Topic

Movimiento

#dotNetSpain2016

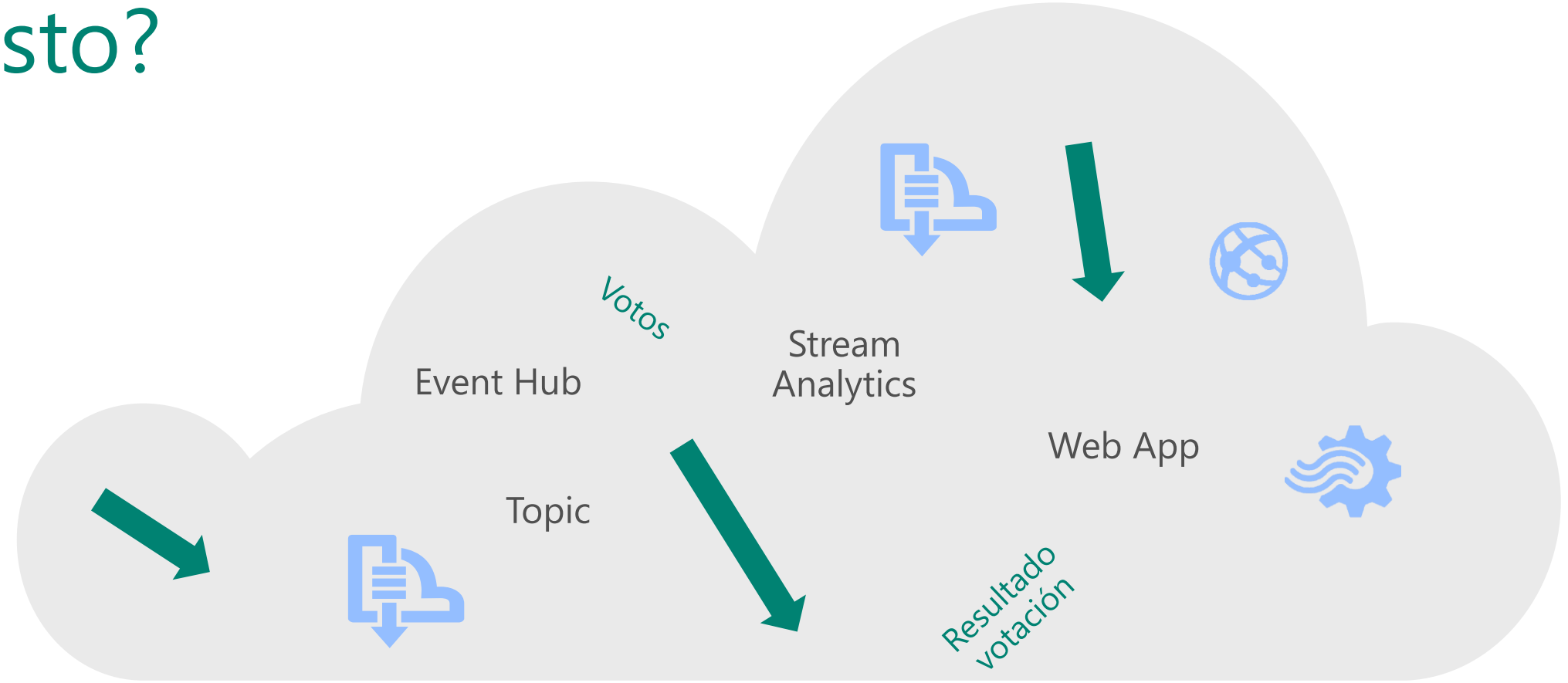
Service Bus Topic





#dotNetSpain2016

La teoría está muy bien pero….¿Cómo monto todo esto?

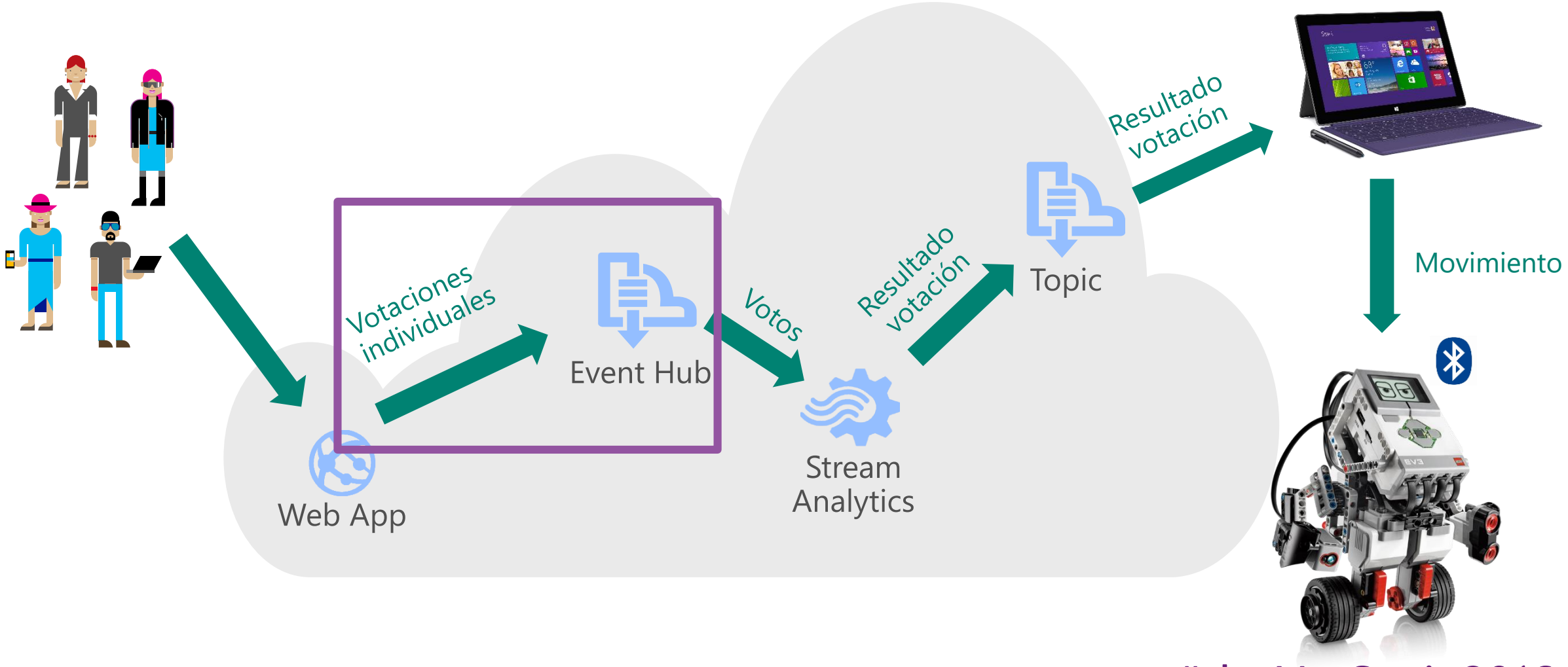


DEMO

Ya tenemos nuestra arquitectura
¿Y ahora?

¡A programar!

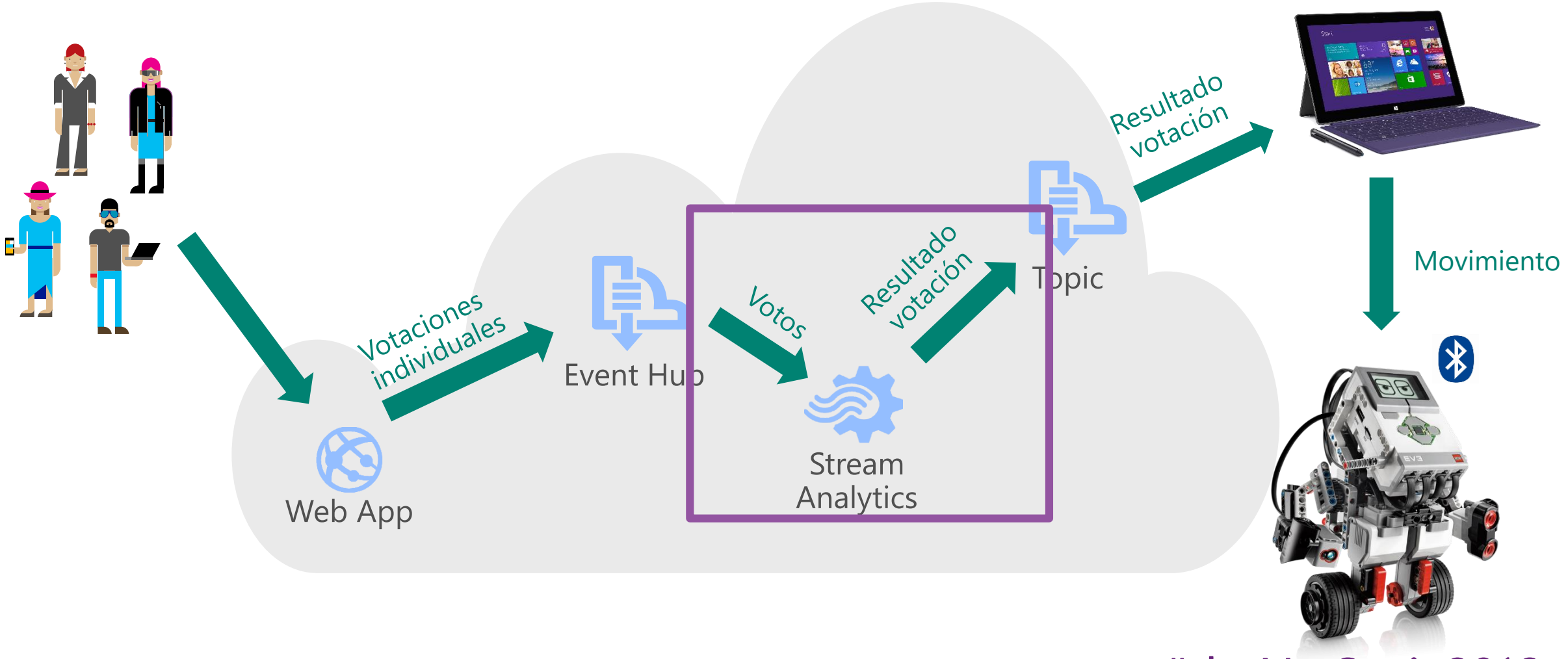
Enviando eventos a Event Hub



Enviando eventos a Event Hub

```
EventHubData ehd = new EventHubData() { Movement = movement, EntryTime= DateTime.UtcNow};

var serializedString = JsonConvert.SerializeObject(ehd);
EventData data = new EventData(Encoding.UTF8.GetBytes(serializedString))
{
    PartitionKey = "0"
};

EventHubClient client = EventHubClient.CreateFromConnectionString(ConfigurationManager.AppSettings["ConnectionString"],
        "legodemoeventhub");

await client.SendAsync(data);
```

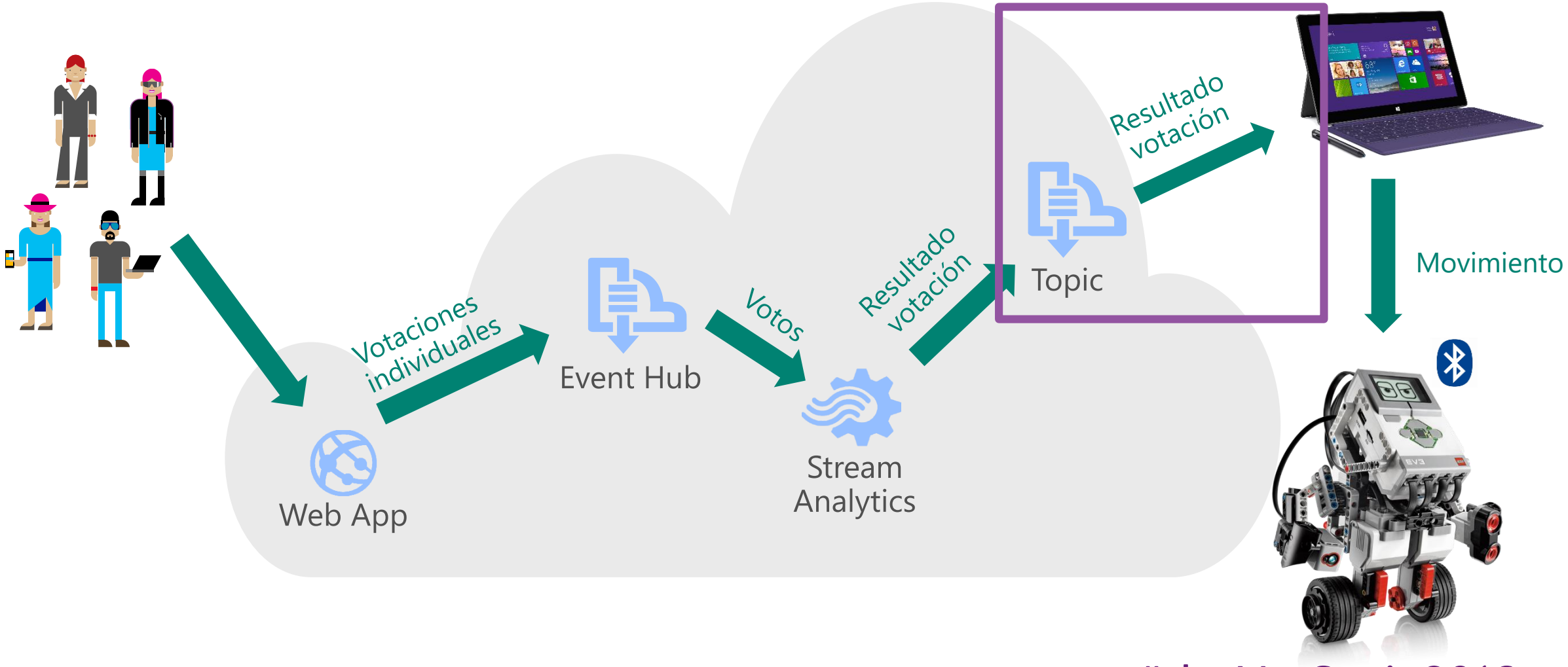
Analizando los datos con Stream Analytics



Analizando los datos con Stream Analytics

```
1  with cnt as (
2      SELECT
3          Movement, Count(*) as votes
4
5      FROM
6          Input TIMESTAMP BY EntryTime
7
8          GROUP BY Movement, TumblingWindow(second,3)
9  ),
10 cnt2 as (
11  select max(votes) as votes from cnt group by TumblingWindow(second,3))
12
13 select SYSTEM.TIMESTAMP as time ,cnt.movement, cnt.votes
14 from cnt
15 join cnt2 on cnt.votes=cnt2.votes and DateDiff(ss,cnt,cnt2) between 0 and 3
16
```

Recibiendo eventos del Topic



Recibiendo eventos del Topic

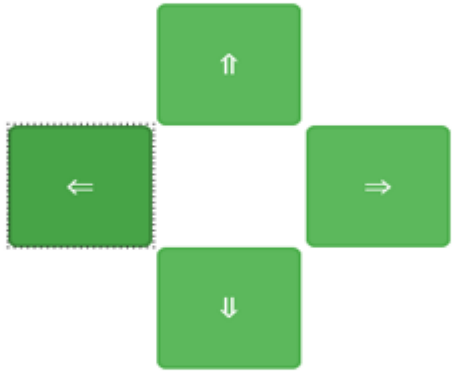
```
var subscriptionClient = SubscriptionClient.CreateFromConnectionString(connectionString, topicPath, subscriptionName);

var brokeredMessage = await subscriptionClient.ReceiveAsync(TimeSpan.FromSeconds(1));

var message = brokeredMessage.GetBody<string>();

var movement = JsonConvert.DeserializeObject<EventHubVotes>(message);
```

¿Jugamos?

http://legoev3web.azurewebsites.net/

¡¡Controla nuestro Robot LEGO!!

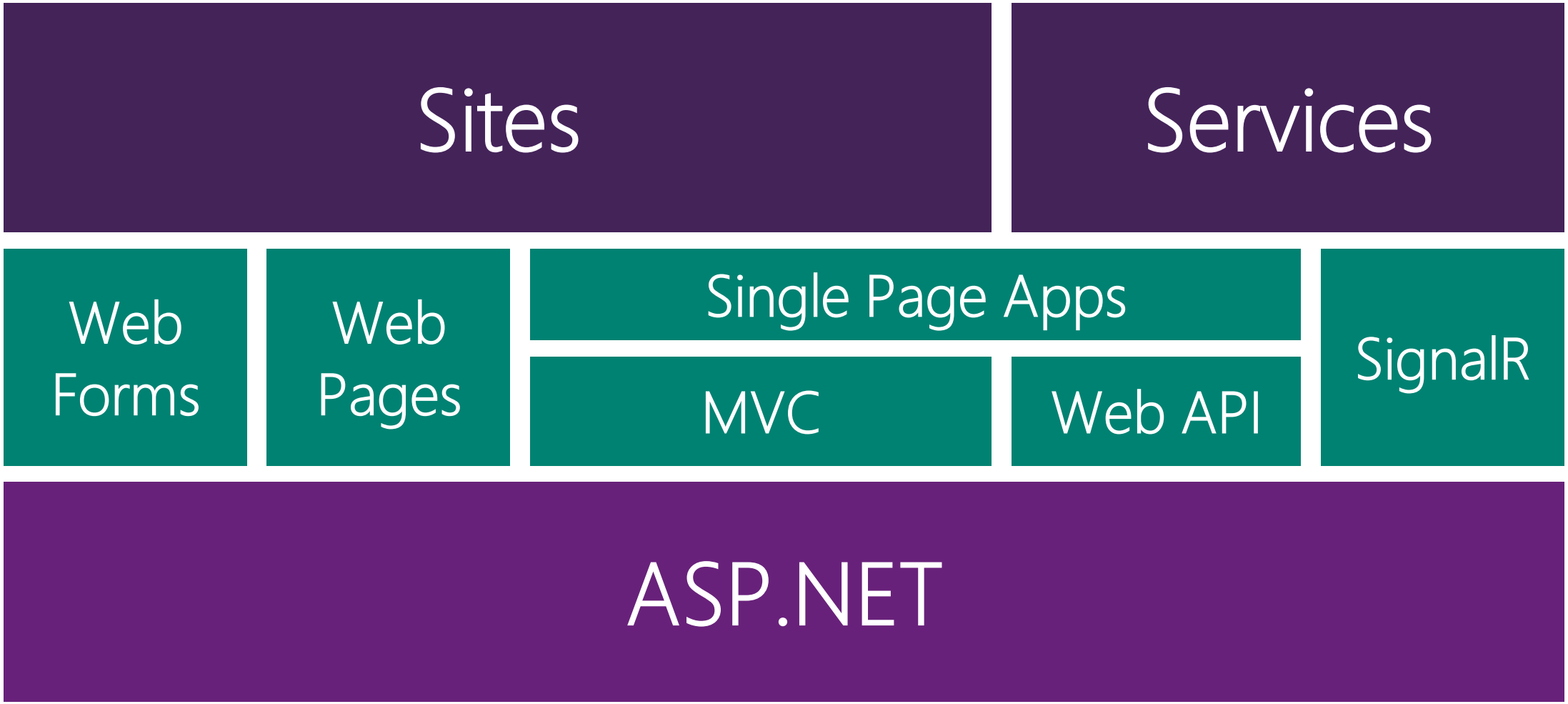
Vota la dirección hacia la que quieres dirigir el robot



Votaciones

Izquierda	Adelante	Derecha	Atrás
14	24	37	21

Funcionamiento ASP.NET



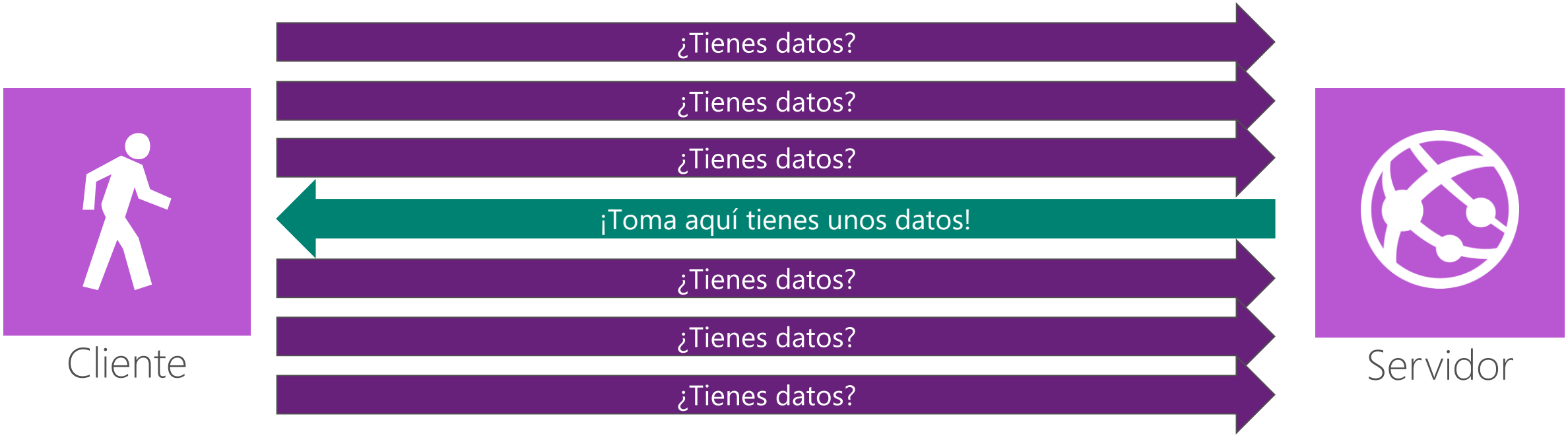
¿Qué es SignalR?

SignalR es una serie de abstracciones alrededor de varios métodos para proporcionar conexiones HTTP persistentes, distribuidos como código open-source.

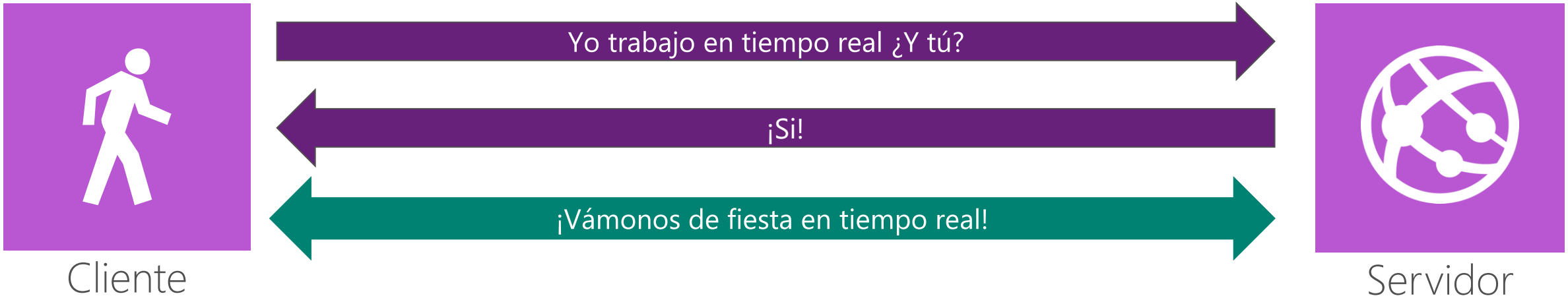
[¿Y esto en castellano?]

SignalR hace las conexiones HTTP en tiempo real tan fáciles que parece magia.

SignalR en Servidores y Clientes Antiguos

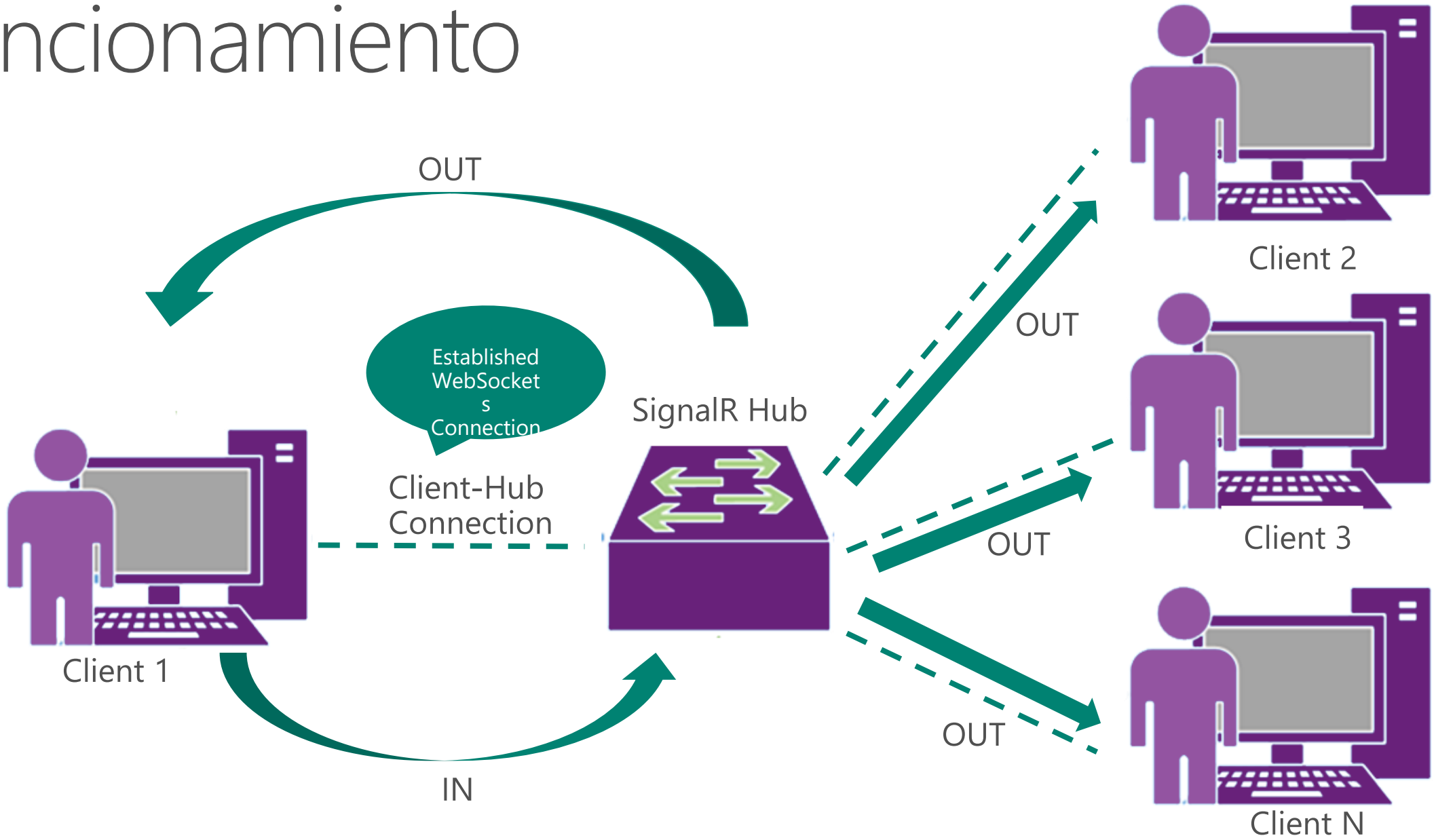


SignalR en Servidores y Clientes Modernos



#dotNetSpain2016

Funcionamiento



#dotNetSpain2016

Puedes encontrar todo el código de las demos en:

http://github.com/isabelcabezasm

Conclusiones

- Lego no es solo para niños
- Jugar todos significa escalar
- Cuando tengas un problema,
  tu solución está en Azure
- Signal R es "magia"

Q&A

http://aka.ms/DOTNETT8S1

#dotNetSpain2016

¡Gracias!

Isabel Cabezas
@isabelcabezasm
Beatriz García
@BeRoces

http://aka.ms/DOTNETT8S1

Microsoft

#dotNetSpain2016