

Why not F#?

Diego Bonilla

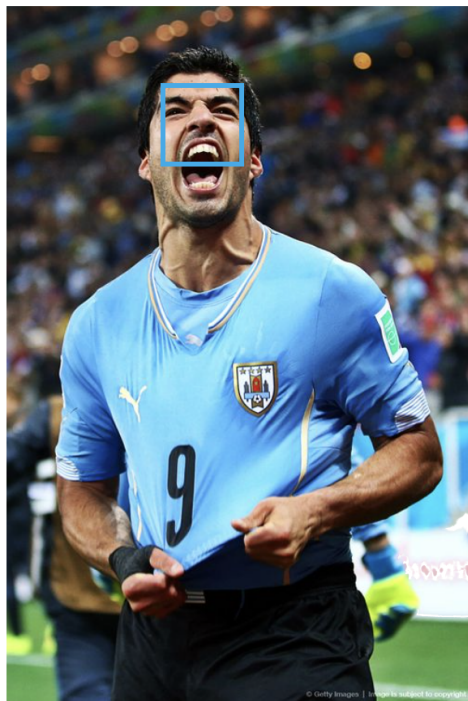
CEO & Co-Founder of Nareia

Organizer @XamarinUY

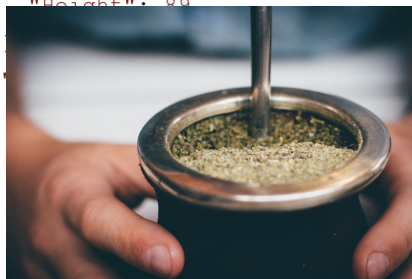
@dbonillanareia



URUGUAY

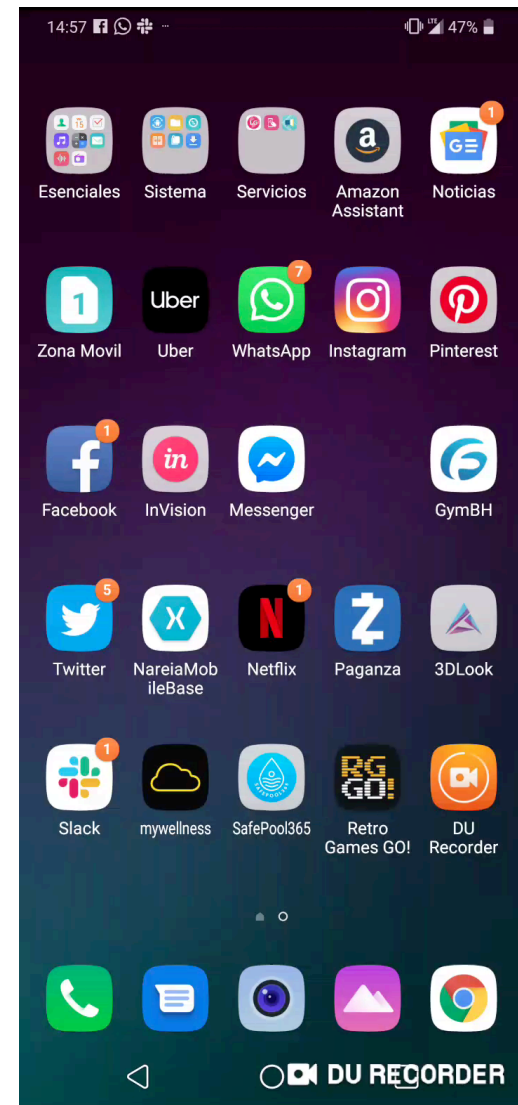


```
{  
  "FaceRectangle": {  
    "Top": 85,  
    "Left": 164,  
    "Width": 89,  
    "Height": 89  
  },  
  "Neutral": 0.000253919832,  
  "Sadness": 0.00151579734,  
  "Surprise": 0.0290504545  
}
```



XAMARINUY

- 2+ years
- meetup.com/xamarinuy
- 367 monkey devs
- github.com/xamarinuy
- Slack
- @xamarinUY
- Retro Games GO!





MOTIVATION

- Maths
- Functional programming
 - Haskell
- F# exists, but **why** do we hear so little about it?
 - Education
 - Offices
 - Community
- You've got some F# in my C#
 - Func
 - Readonly
 - async
 - LINQ



WHY NO ONE USES FUNCTIONAL LANGUAGES (1998)

- Philip Wadler, 1998
 - Theory of Functional programming
 - Design of Haskell
- Reasons
 - Compatibility
 - Libraries
 - Portability
 - Availability
 - Packageability
 - Tools
 - Training
 - Popularity



DON SYME

- 1998 Java, Ocaml, Scheme, Lisp, Haskell, COM, TclTk, Pizza, GJ, MLj
 - Microsoft Research & .NET
- C# 1.0 released!
 - Improve C# or a new language?
 - 1998 - 2005: design .NET Generics & C#/VB Generics
 - But... It's not enough
 - F# was born (2005)!
 - OCaml for .NET?



- Multi-paradigm
 - Functional
 - Imperative
 - Object-oriented
 - Metaprogramming
 - Reflective
 - Concurrent



Version	Date	Platforms	Runtime/IDE
F# 1.x	May 2005	Windows	.NET
F# 2.0	April 2010	Linux, macOS	Mono/Visual Studio
F# 3.0	August 2012	JavaScript, GPU	
F# 3.1	October 2013		
F# 4.0	July 2015		
F# 4.1	March 2017		.NET Core

F# GOES OPEN SOURCE



- Posted on 11 Nov 2010

F# is a fascinating language, but I had not really spent much time with it as we could not really distribute it as an open source compiler limiting its usefulness in the Linux and Mac worlds. Now F# can become just another language that developers can use.

F# supports asynchronous programming today, and it was the inspiration for C# 5.0's async support. There is no need to wait for C# 5 to come out, you can start using async workflows today with F# everywhere.

Miguel de Icaza



FREEDOM OF F#

- Fsharp.org
 - 2012
 - Tomas Petricek, Phillip Trelford and Don Syme
- Open source and Cross platform
- Github, 2015



EVOLUTION

1. 2005
2. 2010 Units of measure, Asynchronous programming
3. 2012 Type Providers, LINQ
4. 2016 Community driven



CHARACTERISTICS

- Indentation & whitespaces
- Immutability by default
- Type providers (the information revolution)
 - SQL
 - JSON
 - CSV
 - Hadoop/Hive
 - R
- Units of measure
- Active patterns
- Async
- GPU



F# VS C# IN XAMARIN

- Same CLR
- Functional-first programming vs OO-first programming
- Learning curve: teaching, dev team
- Some mobile *ilities* that we should consider:
 - Flexibility
 - Maintainability
 - Without bugs?
 - Scalability
 - Usability
 - Testability
 - Reusability
 - Performance
 - Benchmarks are tricky



BENCHMARK

- App
 - Add monkeys / List
 - Get monkeys / Rest service
 - Get photos / Rest service
- BenchmarkDotNet
- <https://github.com/dbonillanareia/fsharp Xamarin/>



BENCHMARK

Code!

RESULTS

Method	N	Mean C# (us)	Mean F# (us)	Winner
AddMonkeys	100	10.91	11.177	-
AddMonkeys alternative f#			10.305	-
GetMonkeys	10000	122,132.695	342,773.659	C# (1/3 F#)
GetPhotos	100	840,010.85	410,218.192	F# (1/2 C#)

// * Legends *

N : Value of the 'N' parameter
Mean : Arithmetic mean of all measurements
Error : Half of 99.9% confidence interval
StdDev : Standard deviation of all measurements
Rank : Relative position of current benchmark mean among all benchmarks (Arabic style)
1 us : 1 Microsecond (0.000001 sec)

F# STRENGTHS

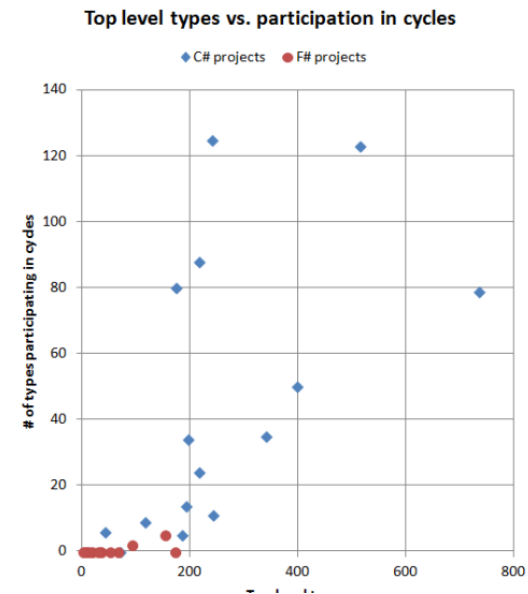
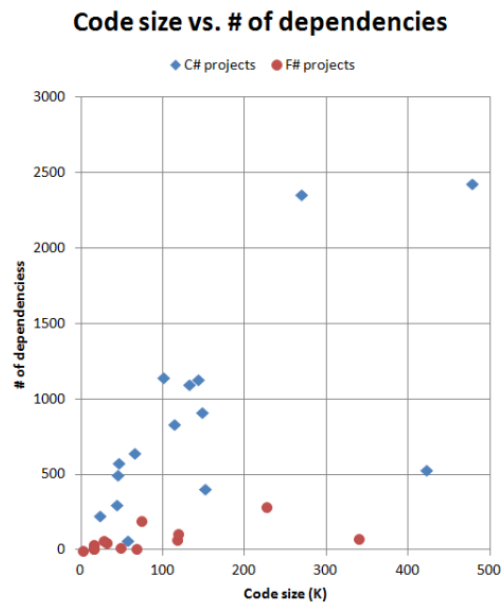
- fsharp.org/testimonials

I have now delivered three business critical projects written in F#. I am still waiting for the first bug to come in. This is not the case with the C# projects I have delivered. I will continue to monitor and report on this. It might be that I am just on a lucky streak, but I suspect that the clarity and concision of F# code contributes greatly to its correctness.

Simon Cousins, Energy Sector

- Predictability
- Circularity
- Lines of code
- Readability
 - Clean code

F# STRENGTHS



Reference: Don Syme "The F# Path to Relaxation"

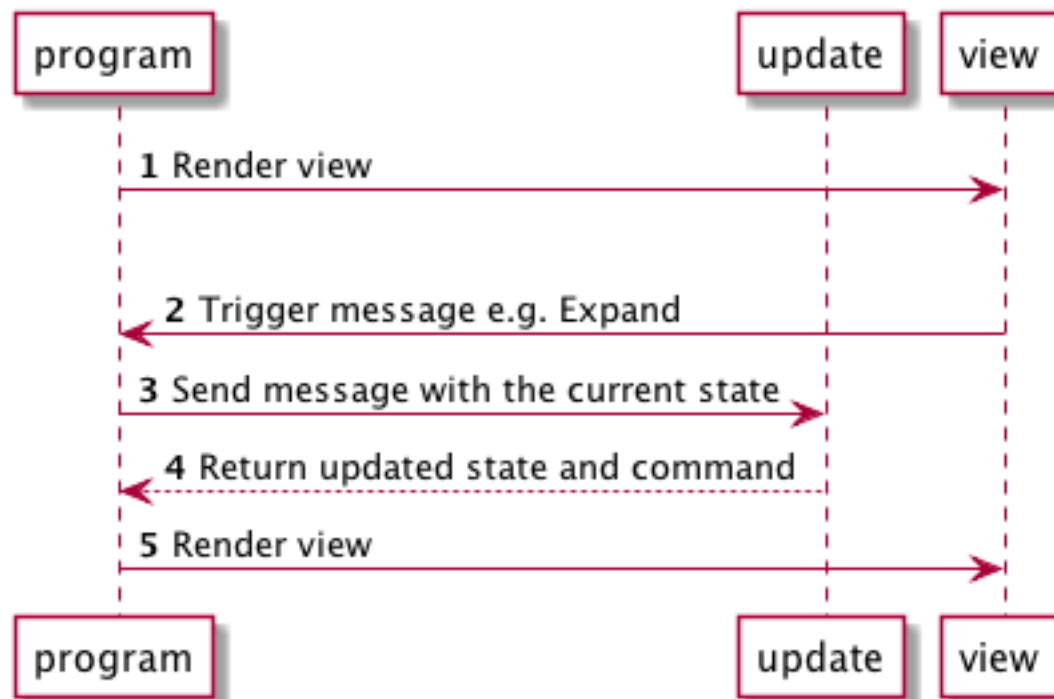


F# WEAKNESSES

- User Interface?

F# + XAMARIN

Elmish (Half)





F# + XAMARIN

Code!



HOW TO START?

- Mindset: thinking functionally
- “a helpful link for your manager”
- fsharp.org
- <https://elmish.github.io/elmish/>



FINAL REFLECTION

A matter of faith

Diego Bonilla
@dbonillanareia
github.com/dbonillanareia/fsharpXamarin

