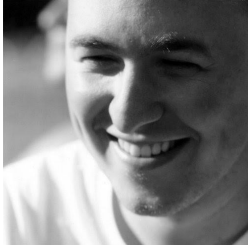


SVILUPPO web

ReactJS e ExtJS a confronto

Speaker: **marco** BreveGLieri



marco
BreveGLIERi
DEVELOPER

www.compilaquindiva.com

Delphi Day 2016
 ***wintech italia***

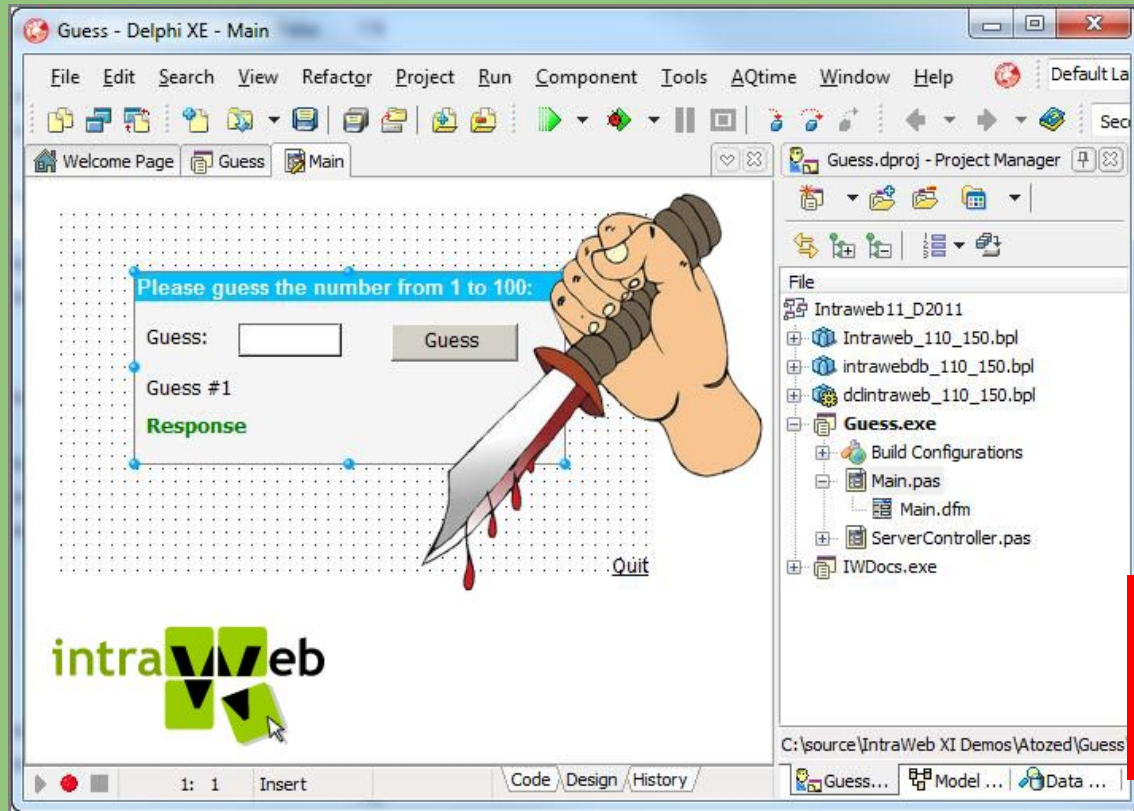
AGENDA

- Le applicazioni Web
- Il linguaggio JavaScript
- Introduzione alle SPA
- La libreria ReactJS
- Dietro le quinte di React
- Backend in Delphi

Le applicazioni web

Soluzioni per applicazioni Web a
confronto

APPLICAZIONI WEB?



NO!

WEB SERVER “FAT” APP

Soluzioni basate su

- ❖ IntraWeb
- ❖ ASP.NET Web Forms

sono applicazioni Web (Server) “Fat”
e non sono più una buona idea. 🙄

WEB SERVER “FAT” APP

PREGI 

- Mascherano la piattaforma (HTTP)
- Facili per sviluppatori abituati al desktop
- Favoriscono lo sviluppo rapido (RAD)
- Sono ideali per la “prototipazione”

WEB SERVER “FAT” APP

DIFETTI

- Mascherano la piattaforma (HTTP)
- Stato della pagina “ingombrante”
- Consumo banda, memoria e CPU server
- Scalabilità generale ridotta
- Separazione task designer/developer improbabile
- Unit/Integration Test ardui

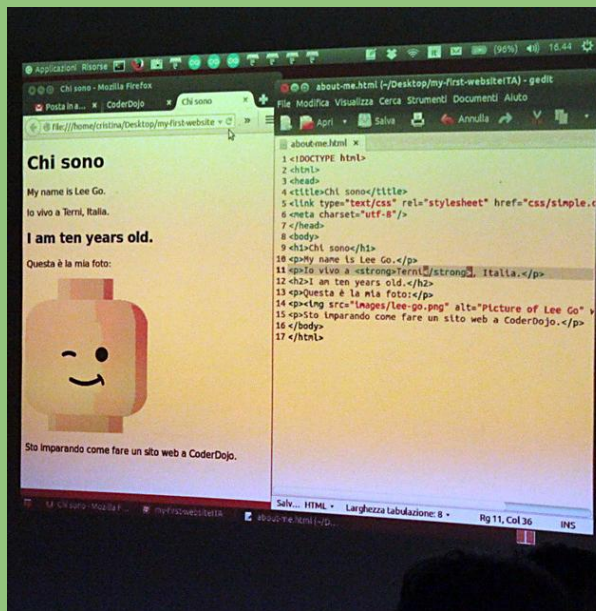
APPLICAZIONI WEB moderne?

Si basano su standard del Web

- HTML(5)
- CSS(3)
- JavaScript



LINGUAGGI STANDARD: SONO DIFFICILI?

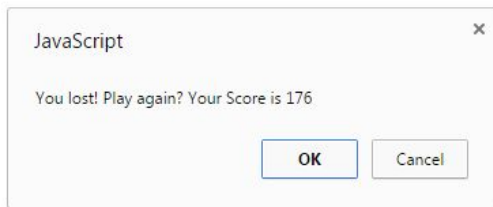


CoderDojo

IL LINGUAGGIO JavaScript

Introduzione a JavaScript e al DOM

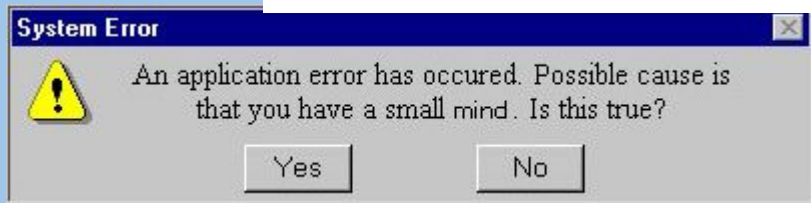
ALL'INIZIO era...



Love Tester

Bill Clinton + Monica Lewinsky = 76.5%

Calculate!



The page at www. .Info says: x

YOUR COMPUTER HAS BEEN LOCKED!!

The work of your computer has been suspended.
Illegal access has been initiated from your system.

Contact Customer Support Help Line: 1-855-

☐ Prevent this page from creating additional dialogs.

OK

OGGI E'....



In evoluzione

Il linguaggio JavaScript vive oggi una “seconda giovinezza”. 🧒

- Supporto esteso da parte dei browser
- Nuovi standard (es. ES6) in arrivo
- Output di linguaggi più evoluti come TypeScript, CoffeeScript e tanti altri

NUOVE API HTML5

- Canvas / SVG
- WebGL
- File API
- Indexed DB
- Media API
- OFFline
- Web Sockets
- Web Workers
- Web Storage
- Geo-location API
- Fullscreen support
- ...e il sempre caro
DOM 😊

IL "DOM"

Il DOM (*Document Object Model*)

- È un modello a oggetti
- È formato da una gerarchia di nodi
- Rappresenta gli elementi della pagina Web
- Fornisce funzionalità per modificare gli elementi, modificarli, eliminarli o crearne di nuovi

demo time



JQUERY: un Passo in +

JQuery è una libreria che

- Semplifica l'accesso al DOM
- Riutilizza le espressioni CSS
- Fornisce “feature” aggiuntive e uniformi
- Gestisce le incompatibilità tra i browser

...tuttavia, non è ancora abbastanza. 🤖

demo time



INTRODUZIONE ALLE "SPA"

Single Page Application

SPA

SPA (*Single Page Application*)

- Applicazione Web in una sola pagina
- Mima le caratteristiche delle app desktop
- Esperienza utente più fluida
- Non ricarica la pagina nel browser
- L'interfaccia utente si aggiorna dinamicamente a seguito di un'azione
- Trasferimento ottimizzato di dati e risorse

Per rendere l'idea

```
<!DOCTYPE html>

<html>

<head>

  <meta charset="UTF-8">

  <title>Single Page Application</title>

  <link rel="stylesheet" href="style.css" />

</head>

<body>

  <div id="container">

    <!-- Qui viene generata l'applicazione -->

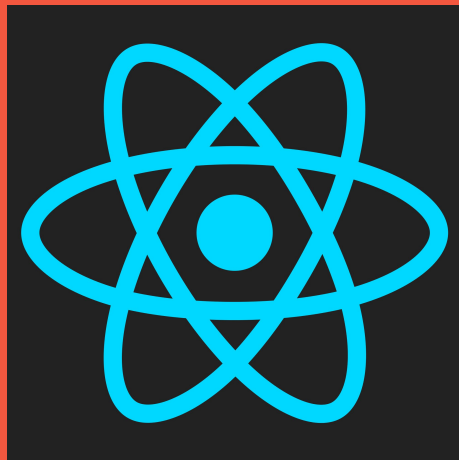
  </div>

  <script src="framework.js"></script>

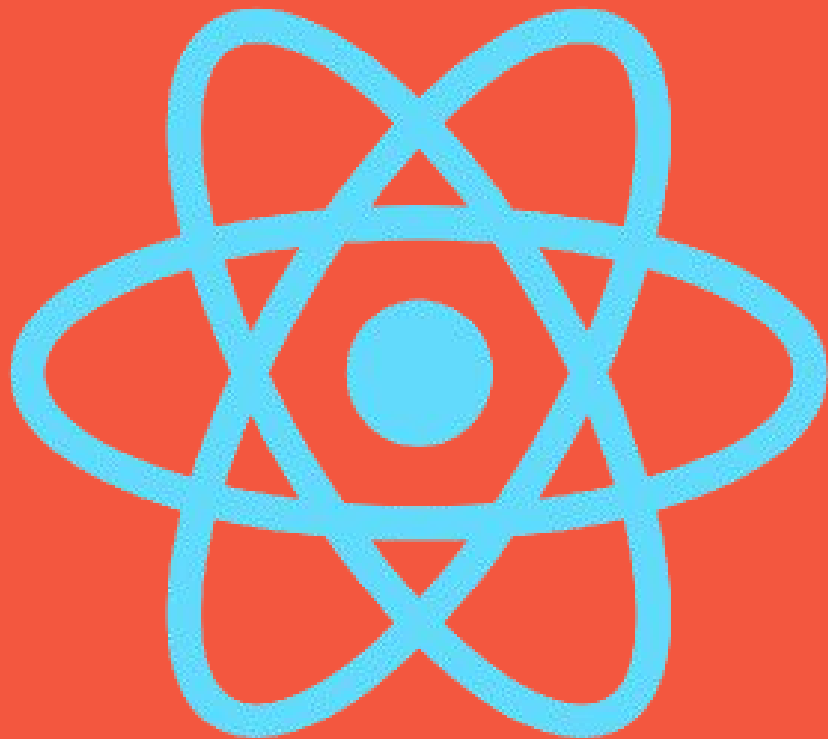
</body>

</html>
```

LIBrerie e Framework



Let's "REACT"!



React

ReactJS

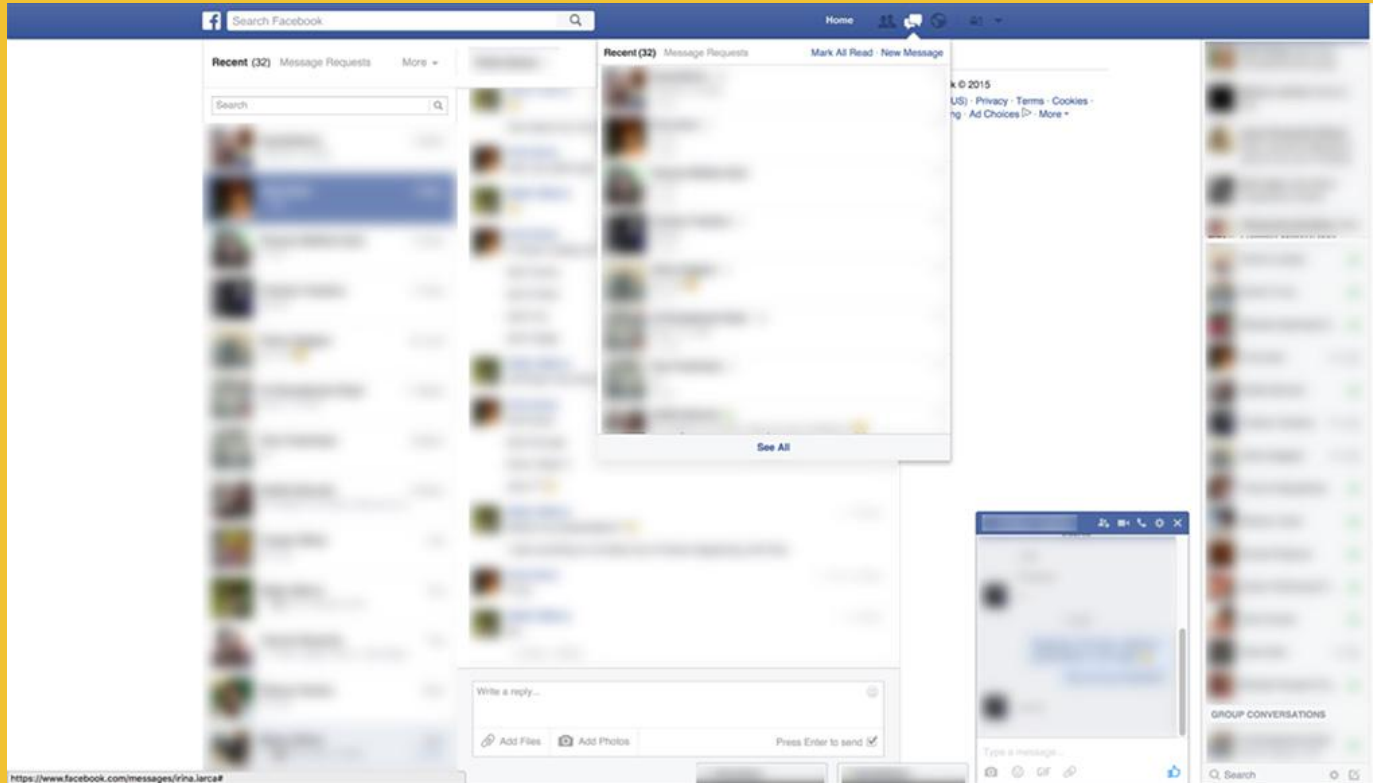
La libreria ReactJS di Facebook™ 

REACTJS

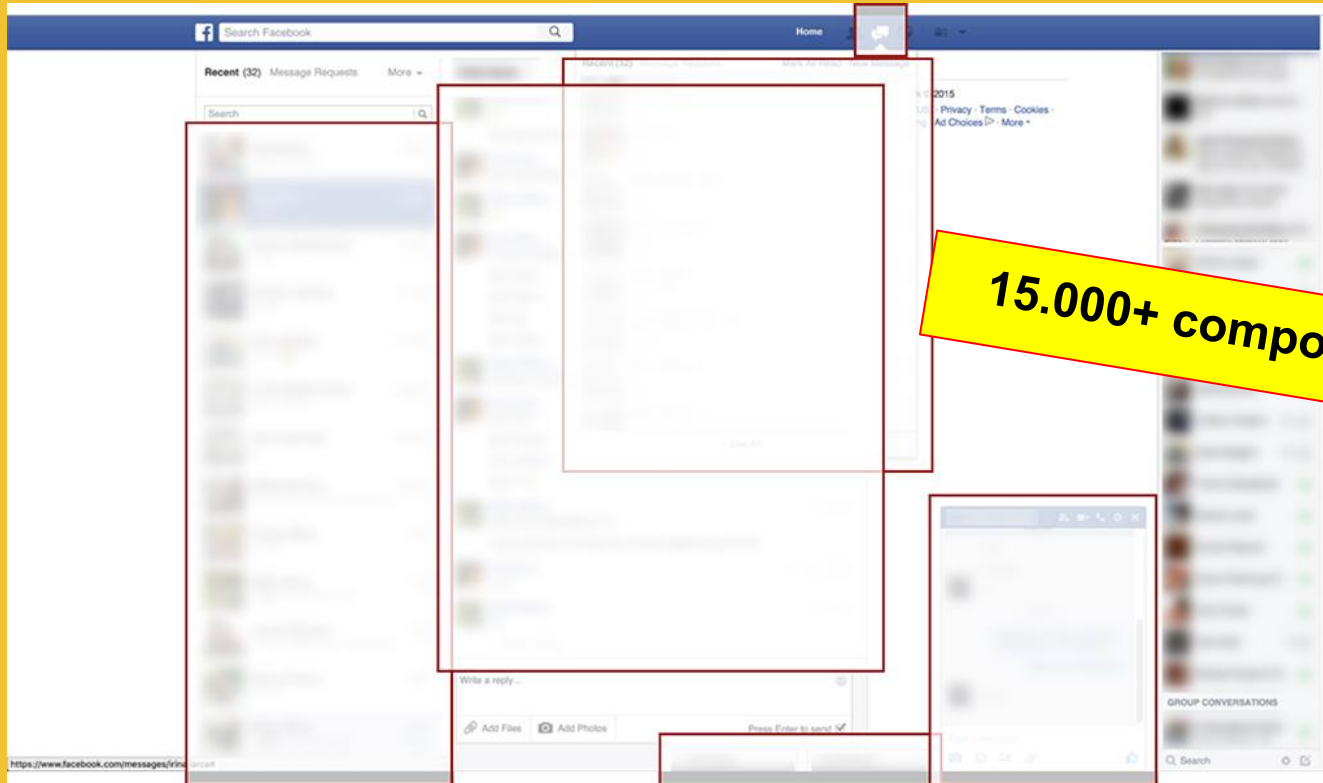
- Libreria JavaScript per le interfacce utente
- Creata da Facebook (anche per Instagram)
- Nel modello MVC gestisce solo la “View”
- Celebre per le sue elevate prestazioni
- Basato su “componenti”, proprietà e stato
- Approccio dichiarativo (*) (*) usando JSX

 <https://facebook.github.io/react/>

FACEBOOK "case study"



FACEBOOK "case study"



15.000+ componenti!

componenti

- Incapsulati
- Riutilizzabili
- Componibili
- Facili da disegnare *

(*) almeno in React

REACTJS VS...

JQuery

- Programmazione imperativa
- Necessità di attribuire ID agli elementi
- Tendenza “event driven”
- Rischio concreto di “zuppa-code”

AngularJS (ver. 1.x)

- Oscillante curva di apprendimento (ma framework completo)
- “Separation of Responsibility” (invece di “...of Concerns”)
- Proliferazione di “scope” e direttive

EDITOR e TOOL

- Notepad++
- Sublime Text
- Atom.IO
- Brackets
- Visual Studio Code
- JetBrains
WebStorm
- Visual Studio
Professional

...con vari plugin 

demo time



La sintassi "JSX"

- Facilita la scrittura di template HTML
- Sostituisce chiamate a funzioni React
- Richiede la presenza di un "compilatore" (detto "transpiler")
- Definisce gli elementi del Virtual DOM *

(*) ne parleremo a breve... 

Proprietà e Stato

Proprietà

- Definiscono valori immutabili del componente
- Utili per inizializzazione
- Si leggono da `this.props`

Stato

- Definisce valori dello stato attuale del componente
- Si modifica per aggiornare l'interfaccia
- Si leggono i valori da `this.state`

demo time



DIETRO Le QUINTE

Il Virtual DOM... e oltre !

I LIMITI DEL DOM

- Non è possibile modificare il codice
- E' una “black box” fornita dal browser
- Non si può ottimizzare in prestazioni
- Non si può specializzare in base all'uso
- Impone uno stile “imperativo” di codifica
- Produce codice meno manutenibile
- Rappresenta un “collo di bottiglia”

IL "VIRTUAL DOM"

- Rappresentazione virtuale della struttura della pagina in memoria
- Riproduce il DOM reale della pagina Web
- Viene usato da React per applicare le modifiche agli elementi della pagina
- Attraverso l'analisi delle differenze con il DOM reale, aggiorna la pagina...
...nel modo più veloce possibile! ⚡⚡⚡

GLI ELEMENTI DEL VDOM

- **ReactNode**

rappresenta un singolo “nodo” del VDOM

- **ReactElement**: rappresenta un elemento HTML
- **ReactText**: rappresenta una porzione di testo

- **ReactFragment**

- **ReactNode[]**: è un array di nodi del Virtual DOM

creare elementi

I parametri della funzione **createElement()**

```
ReactDOM createElement(  
  string/ReactClass type,  
  [object props],  
  [children ...]  
)
```


demo time



component lifecycle

Le funzioni principali del ciclo di vita di un componente:

- *render()*
- *getInitialState()*
- *componentWillMount()*
- *componentDidMount()*
- *componentWillUnmount()*

...e altri meno utilizzati.

demo time



ASSEMBLARE IL TUTTO

Ricetta per combinare gli elementi visti sino ad ora:

- Creare la pagina di struttura della SPA
- Importare i framework preferiti (incluso React!)
- Creare script separati per i componenti
- Richiamare la funzione `ReactDOM.render()`

Vediamo ulteriori esempi! 👍

demo time



Backend

Comunicare con il server

E DELPHI?

- Ottimo per la creazione di un backend
- Diverse librerie disponibili per creare API
 - ◆ DataSnap / EMS
 - ◆ DelphiMVCFramework / MARS
 - ◆ Indy Components
- Versatile nel collegamento a qualsiasi DB
- Si può invocare dall'applicazione React *

(*) con la libreria che si preferisce per il colloquio con endpoint HTTP/REST

demo time



ALTro?

→ **Integrazione con altri framework**

◆ Flux, Redux, ...

→ **Altre implementazioni?**

◆ React Native

→ **Possibilità ed estensioni?**

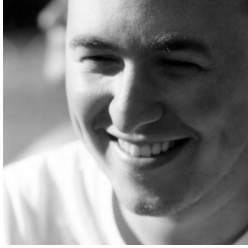
◆ Rendering sul server (con NodeJS)

→ **Migliorare il codice?**

◆ Uso del transpiler “Babel”, pre-compiling server

Q & A

Dubbi, richieste, domande?

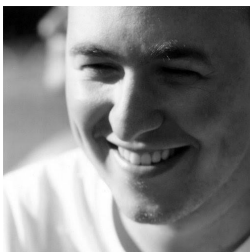


marco
BreveGLieri
DeVELOPER

www.compilaquindiva.com



Delphi Succinctly: e-book di 100 pagine in lingua inglese, scaricabile gratuitamente da Syncfusion Inc.

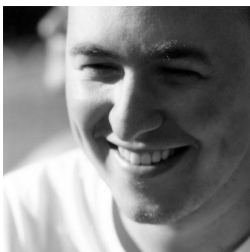


marco
BreveGLieri
DeVELOPER

www.compilaquindiva.com

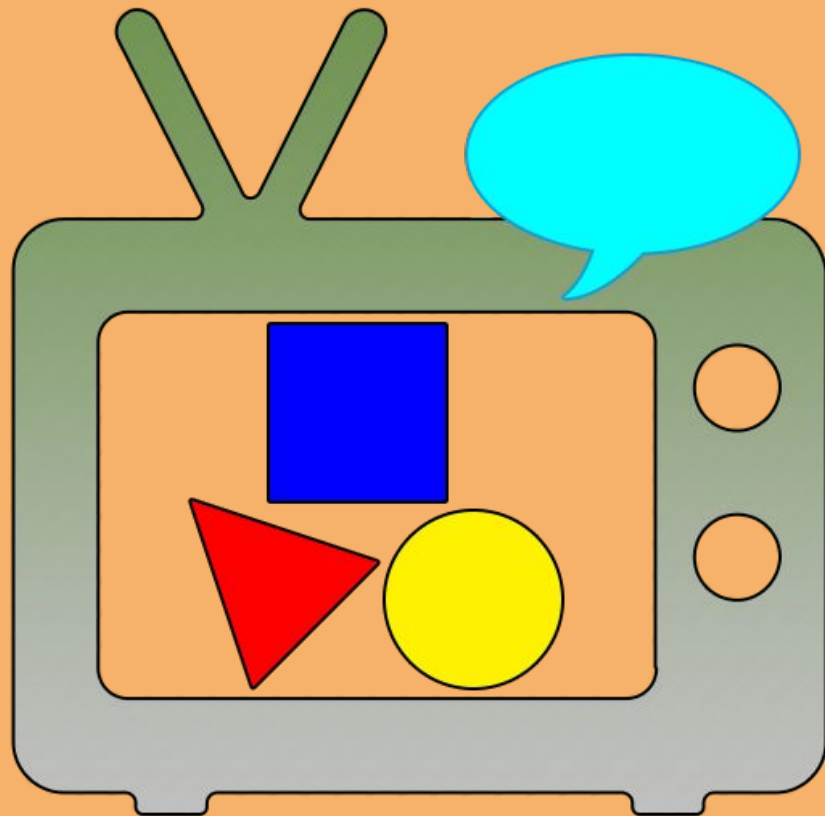


Delphi Podcast - www.delhipodcast.com
Il primo podcast italiano dedicato a Delphi.



marco
BreveGLieri
deVELOPER

www.compilaquindiva.com



@ComponentTales - <https://twitter.com/ComponentTales>
Scopri cosa fanno i componenti quando chiudi l'IDE. 🐞

grazie a TUTTI!

