

Ruby on Rails

Lugano, 29.03.2014



railsgirlsticino@gmail.com | [@RailsGirls_TI](https://twitter.com/RailsGirls_TI) | WWW.RAILSGIRLS.COM/TICINO



- ◆ INTRODUZIONE
- ◆ PRINCIPI BASE
- ◆ PRINICIPI BASE – ROR
- ◆ SVILUPPIAMO
- ◆ CONCLUSIONI



01. Introduzione

Introduzione – La Storia



- ◆ Il linguaggio di programmazione Ruby è stato progettato nel 1993 dal giapponese Yukihiro Matsumoto

キヤ kya	キュ kyu	キョ kyo
シャ sha	シュ shu	ショ sho
チャ cha	チュ chu	チョ cho
ニヤ nya	ニュ nyu	ニョ nyo
ヒヤ hya	ヒュ hyu	ヒョ hyo
ミヤ mya	ミュ myu	ミョ myo
リヤ rya	リュ ryu	リョ ryo
ギヤ gya	ギュ gyu	ギョ gyo
ジャ ja	ジュ ju	ジョ jo
ビヤ bya	ビュ byu	ビョ byo
ピヤ pya	ピュ pyu	ピョ pyo



- ◆ Ma inizialmente non ebbe molto successo!
 - Per più di un decennio non prese mai il decollo
 - Inizialmente la documentazione era esclusivamente in giapponese!!!



- ◆ L'anno di svolta fu il 2006
 - Anno in cui si afferma il framework Ruby on Rails

Introduzione – Alcune caratteristiche



Ruby è un linguaggio:

- ◆ open source
- ◆ general purpose
- ◆ interpretato
- ◆ orientato agli oggetti

Possiamo definirlo come un linguaggio che ha acquisito tanto da parte di altri linguaggi come: Smalltalk, Perl, Python, C, C++, PHP, Python...

<http://www.ruby-lang.org/en/documentation/ruby-from-other-languages/>

Infatti uno dei primi slogan era il seguente:

(Smalltalk + Perl) / 2

Introduzione – Cos'è Ruby On Rails (ROR) ?



RUBY ON RAILS IS

“... an Open-source web framework that’s optimized for programmer happiness and sustainable productivity. It lets you write beautiful code by favoring convention over configuration.”

“... a breakthrough in lowering the barriers of entry to programming. Powerful web applications that formerly might have taken weeks or months to develop can be produced in a matter of days.”

(fonte: <http://www.rubyonrails.org>)

- 💎 Ruby on Rails oppure RoR , è un framework open source MVC (Model View Controller) basato su Ruby.
- 💎 Con Ruby on Rails , è possibile sviluppare applicazioni in modo semplice diminuendo la percentuale di codice che solitamente va a ripetersi nelle applicazioni.
- 💎 Il pattern Model View Controller infatti è una pratica di programmazione che semplifica la separazione tra presentazione dei dati, logica della app e contenuti.
- 💎 Le app sviluppate in Ruby on Rails sono particolarmente indicate per progetti dinamici, flessibili che necessitano aggiornamenti continui o ampliamenti futuri.

Introduzione - Perché Ruby On Rails

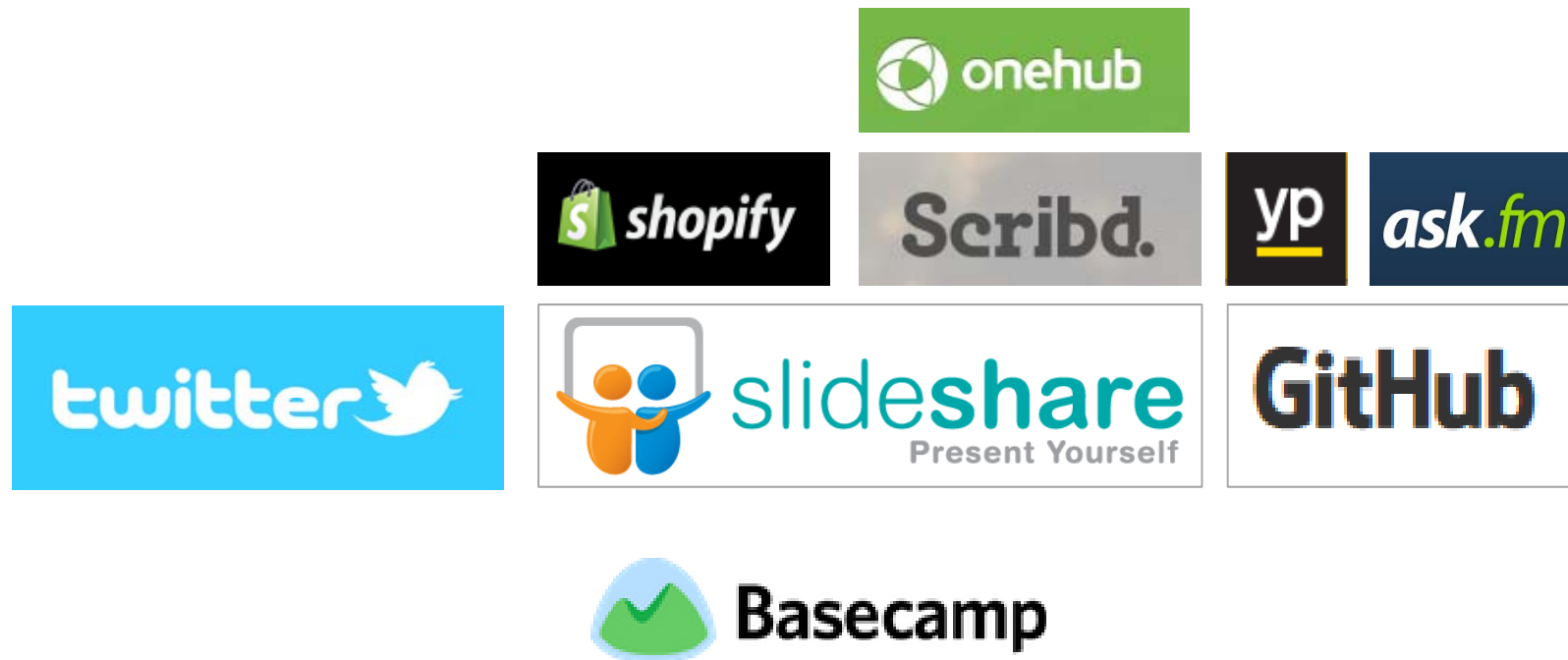
Punti di forza di RoR



PERCHÈ SCEGLIERE RUBY ON RAILS PER LO SVILUPPO?

- 💎 Semplice
- 💎 Intuitivo
- 💎 Scalabile
- 💎 Veloce
- 💎 Diminuisce la complessità del codice
- 💎 Il codice è autoesplicativo
- 💎 Eliminando le fasi più noiose ci permette di concentrarci su ciò che conta veramente: l'idea dello user / il business del cliente.

Introduzione – Websites On Rails



Twitter, SlideShare, GitHub, Basecamp, Shopify, Scribd, OneHub, Yellow Page, Ask.fm, Cookpad, Hulu, CrunchBase, Zendesk, MedHelp, Freckle, CrazyEgg, 43Things.....

Introduzione – Chi usa Rails





02. Principi base

Programmazione Orientata agli Oggetti (OOP)



Definizione

Questo permette di raggruppare in un porzione circoscritta (classe) del codice sorgente la dichiarazione delle strutture dati e delle procedure che operano su di esse.

- ◆ Nella programmazione orientata agli oggetti (OOP, Object Oriented Programming) un oggetto può essere visto come un contenitore di dati dotato di attributi (dati) dotato di una serie di metodi (procedure) secondo quanto dichiarato dalle rispettive classi

- ◆ L'oggetto è una istanza di una classe
 - Esso è dotato di tutti gli attributi e i metodi definiti dalla classe.

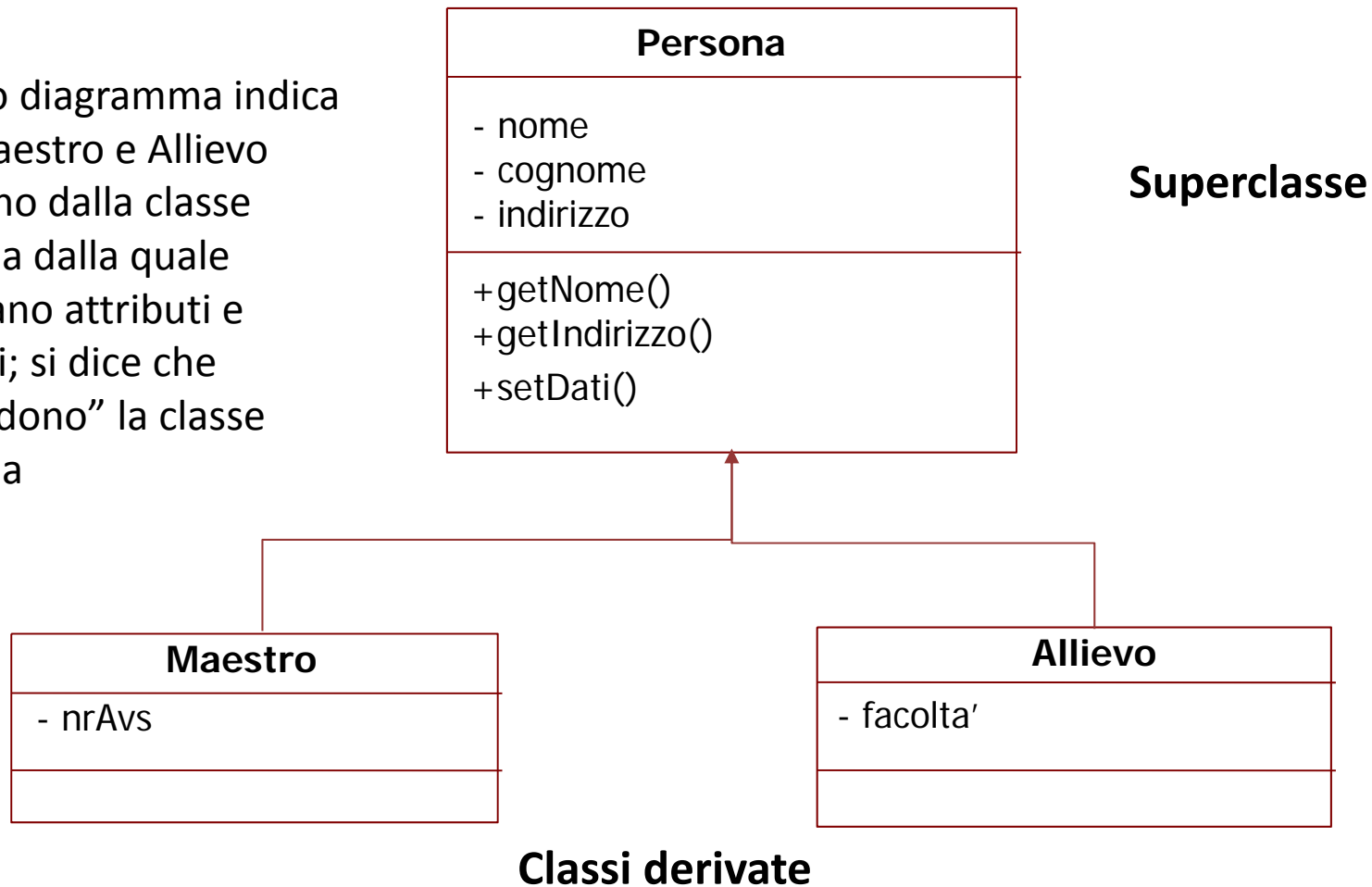
- ◆ Caratteristiche :
 - ◆ Ereditarietà
 - ◆ Polimorfismo
 - ◆ Incapsulamento

Programmazione Orientata agli Oggetti (OOP)

Eredità



Questo diagramma indica che Maestro e Allievo derivano dalla classe Persona dalla quale ereditano attributi e metodi; si dice che “estendono” la classe Persona



Programmazione Orientata agli Oggetti (OOP)

Esempio



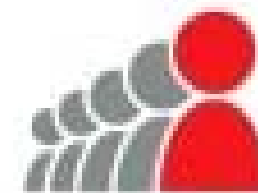
```
class Persona
  attr_accessor :nome, :cognome, :indirizzo

  def initialize (nome, cognome, indirizzo)
    @nome = nome
    @cognome = cognome
    @indirizzo = indirizzo
  end
end
```



Oggetto

Classe





03. Principi base - ROR

MVC

Il modello Model View Controller



I dati sono separati (model) dall'interfaccia utente (view)

Model

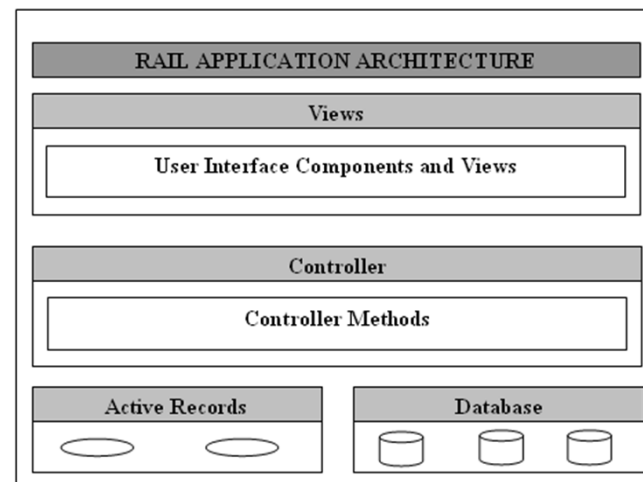
- Mantiene il rapporto tra oggetti e database e gestisce la convalida, l'associazione, le transazioni, e altro ancora.
- Accesso ai dati e alla logica di business
- Indipendente dalla view e dal controller.

View

- Presentazione dei dati e interazione con l'utente
- Accesso in sola lettura al modello

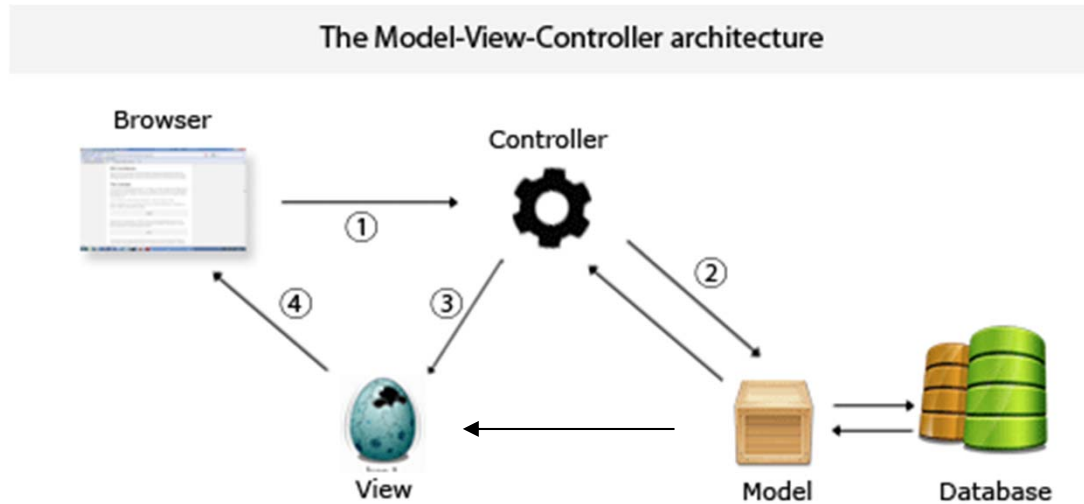
Controller

- È il centro logica dell'applicazione.
- Coordina l'interazione tra l'utente, le viste, e il modello.
- Gestione degli eventi



MVC

Funzionamento



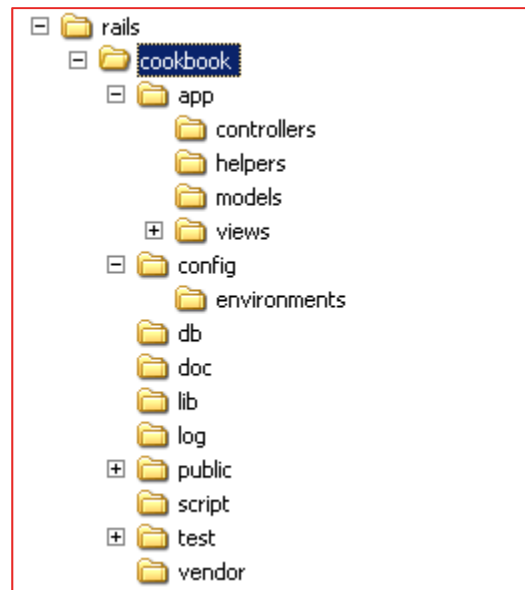
1. Il browser invia le richieste
2. Il controller interagisce con il modello
3. Il controller chiama la vista
4. La vista produce la schermata sul browser

Struttura di un applicazione Ruby on Rails



- ❖ Le applicazioni sviluppate con Rails hanno una peculiarità, ovvero sono tutte organizzate secondo una struttura comune.
- ❖ Questa è una conseguenza del fatto che il comando rails genera una serie di directory e file che forniscono una certa linea guida nello sviluppo, linea che se rispettata permette a Rails di effettuare molte cose automaticamente (ad esempio caricare i file, generarli ed individuarli a runtime e molto altro).
- ❖ Questa struttura comune permette anche di comprendere con semplicità il codice di progetti realizzati da altri, in quanto sono organizzati nella stessa maniera.

❖ Struttura applicazione



Convenzioni sui nomi – Ruby on Rails



Rails preferisce le convenzioni alle configurazioni, e quindi cerca di evitare allo sviluppatore il peso di dover specificare l'associazione ad esempio tra tabelle e classi.

Per l'associazione tra tabelle e classi del modello è sufficiente che le tabelle siano chiamate con il plurale del nome della classe e che siano scritte in minuscolo.

ESEMPIO : La tabella messages sarà dunque mappata sulla classe Message, authors su Author e topics su Topic. Il meccanismo di ActiveRecord che si occupa di questa conversione è molto più intelligente di quel che si potrebbe pensare e ad esempio è in grado di capire che il plurale di "person" è "people" o che il plurale di "status" è "statuses".

Modelli

Tabella	products
File	app/models/product.rb
Classe	Product

Controller

URL	http://../store/list
File	app/controllers/store_controller.rb
Classe	StoreController
Metodo	list
Layout	app/views/layouts/store.html.erb

Viste

URL	http://../store/list
File	app/views/store/list.html.erb (o .builder o .rjs)
Helper	modulo StoreHelper
File	app/helpers/layouts/store_helper.rb



04. Sviluppiamo

<http://guides.railsgirls.com/app/>



05. Conclusioni

Tutorial



<http://railsforzombies.org/>

<http://ruby.railstutorial.org/>

<http://railscasts.com/>

<http://guides.rubyonrails.org/>

Conclusioni



DOMANDE

THANK YOU



<http://railsgirls.com/ticino>

[email] railsgirlsticino@gmail.com

[twitter] [@RailsGirls_TI](https://twitter.com/RailsGirls_TI)

[facebook] [Rails Girls Ticino](https://www.facebook.com/RailsGirlsTicino)
