

Al lavoro con Phoenix

Ricordi di un'estate

di Paolo Montrasio
paolo.montrasio@connettiva.eu

Slide a
connettiva.eu/al-lavoro-con-phoenix

Paolo Montrasio

Architect

Devop

Developer (Ruby, Python, JavaScript, Elixir)

Di ritorno da un viaggio

Ricordi belli

Ricordi spiacevoli

Sorprese piacevoli

Sorprese meno piacevoli

Aneddoti per gli amici

Phoenix come compagno di viaggio

Ricordi belli

Non dà fastidio

Ricordi belli

Non dà fastidio

Si lamenta in modo comprensibile

```
$ iex -S mix
```

```
Erlang/OTP 19 [erts-8.3] [source] [64-bit] [smp:8:8] [async-  
threads:10] [hipe] [kernel-poll:false]
```

```
[error] Postgrex.Protocol (#PID<0.535.0>) failed to connect: **  
(DBConnection.ConnectionError) tcp connect (localhost:9876):  
connection refused - :econnrefused
```

```
... altri 26 errori identici per altri 26 processi ...
```

```
[error] Postgrex.Protocol (#PID<0.522.0>) failed to connect: **  
(DBConnection.ConnectionError) tcp connect (localhost:9876):  
connection refused - :econnrefused
```

```
...
```

```
$ rails c
```

```
Loading development environment (Rails 4.2.3)
```

```
2.2.2 :001 > User.first
```

```
PG::ConnectionBad: could not connect to server: No such file or directory
```

```
Is the server running locally and accepting  
connections on Unix domain socket  
"/var/run/postgresql/.s.PGSQL.5432"?
```


Ricordi belli

Non dà fastidio

Si lamenta in modo comprensibile

Non devi dirgli tutto

Processi in background

Nasconde benissimo tutti i suoi processi

Processi in background

Nasconde benissimo tutti i suoi processi

Non serve installare celery, sidekiq, etc...

```
message = Swoosh.Email.new()  
  |> to({"Somebody", "somebody@example.com"})  
  |> Swoosh.Email.from({"Phoenix", "phoenix@example.com"})  
  |> subject("Welcome")  
  |> render_body("template.html", %{some: "data"})  
  |> render_body("template.text", %{some: "data"})  
Task.start(fn -> deliver(message) end)
```

Processi in background

Nasconde benissimo tutti i suoi processi

Non serve installare celery, sidekiq, etc...

Aggiunta server (es: eventi e notifiche automatiche)

Ricordi belli

Non dà fastidio

Si lamenta in modo comprensibile

Non devi dirgli tutto

Senza se e senza ma

Senza se e senza ma

6 `if` nei file `conf/`, `router.ex`, `endpoint.ex`

1 `if` nei file `test/`

13 `if` nei controller (10 di una lib non nostra)

8 `if` in `views/*helper.ex` (7 in quella lib)

7 `if` negli altri file scritti da noi

Senza se e senza ma

18 nostri + 17 altrui = 35 if

Una applicazione Rails comparabile: 167 if

Senza se e senza ma

end
end
end
end
end
end
end
end
end

}
}
}
}
}
}
}
}
}

Senza se e senza ma

```
defp message(payload = %{"_type" => "Event1"}) do
  ...
defp message(payload = %{"_type" => "Event2"}) do
  ...
defp message(payload = %{"_type" => "Event3"}) do
  ...
```

Senza se e senza ma

```
iex(1)> defmodule Demo do
...(1)>   def x_uguale_y(%{x: a}, %{y: a}), do: :ok
...(1)>   def x_uguale_y(_, _), do: :error
...(1)> end
iex(2)> Demo.x_uguale_y(%{x: 1}, %{y: 2})
:error
iex(3)> Demo.x_uguale_y(%{x: 1}, %{y: 1})
:ok
iex(4)> Demo.x_uguale_y(%{x: 1, y: 2}, %{y: 1, z: 3})
:ok
```

E i ma?

E i ma?

```
iex(1)> ma
warning: variable "ma" does not exist and is being expanded to
"ma()", please use parentheses to remove the ambiguity or change
the variable name
  iex:1

** (CompileError) iex:1: undefined function ma/0
```

Ricordi belli

Non dà fastidio

Si lamenta in modo comprensibile

Non devi dirgli tutto

Senza se e senza ma

Non si fida degli estranei

github.com/smpallen99/coherence

Login / Logout

Registrazione

Invio mail di attivazione

Non ho ricevuto il link di attivazione

Ho perso la password

Plugin per FB / Tw / Google / OAuth

Ricordi meno belli

Ha qualche abitudine strana

Stranezze

web/ e se la mia app non è per il web?

priv/repo perché il db è privato e il codice no?

Rotte plurali, controller singolari

`/posts/8` → `PostController` o `PostsController`

Puntigliosità

```
iex(1)> Model.create(%{name:"Name"})
```

```
** (SyntaxError) iex:1: keyword argument must be  
followed by space after: name:
```

```
iex(2)> Model.create(%{name: "Name"}) # lo spazio :- )
```

Ricordi meno belli

Ha qualche abitudine strana

Certe volte devi ripetergli le cose

Ecto

Campi sia nella migrazione che nel modello

Ecto

Campi sia nella migrazione che nel modello

NO!!!

Ecto

Campi sia nella migrazione che nel modello

Ripetere le migrazioni sul db test dopo i rollback



< pomodori >

L'ottimo è nemico del buono

L'ottimo è nemico del buono

L'ottimo

```
Repo.one(from m in Model, where: m.field == ^field)
```

L'ottimo è nemico del buono

L'ottimo

```
Repo.one(from m in Model, where: m.field == ^field)
```

Il buono

```
Model.get_by(m.field: field)
```

Ecto

Campi sia nella migrazione che nel modello

Ripetere le migrazioni sul db test dopo i rollback

Changeset e DSL faticosissimi

github.com/sheharyarn/ecto_rut

```
Post.all  
# instead of Repo.all(Post)
```

```
Post.get(2)  
# instead of Repo.get(Post, 2)
```

```
Post.insert(title: "New Post", slug: "new-post", category_id: 3)  
# instead of:  
# changeset = Post.changeset(%Post{}, %{title: "New Post", slug:  
"new-post", category_id: 3})  
# Repo.insert(changeset)
```

Alternative e warning

ecto_shortcuts, simile a ecto_rut

<https://elixirforum.com/t/need-code-review/1770/5>

Atlas, alternativo a Ecto (defunto)

 sheharyarn:

I have a few concerns with this project, especially if I might be going against Elixir patterns. I would really appreciate if I could get some feedback on the code before I publish it.

The issue with the approach above is that you are coupling your schema with one particular database and the schema is designed to represent any data source, which may not even be a database in the first place. We explore some of those things here: http://blog.plataformatec.com.br/2016/05/ectos-insert_all-and-schemaless-queries/

My concern with using something such as shortcuts is that people will forget those basic tenets schemas and repositories were built on. So I would rather define modules such as `MyApp.Accounts.get_user(id)`, which is the direction Phoenix is taking, without a need to tie repositories directly into schemas.

sheharyarn 2016-09-25 11:03:10 UTC #6

I've decided to go with the name **Ecto.Rut**

 josevalim:

My concern with using something such as shortcuts is that people will forget those basic tenets schemas and repositories were built on. So I would rather define modules such as `MyApp.Accounts.get_user(id)`, which is the direction Phoenix is taking, without a need to tie repositories directly into schemas.

The intention here is to offer simple wrappers around `Ecto.Repo` methods and simplify common use-cases in Ecto. This can be a means to ease people into the "Ecto way" or just simply reduce code repetition. But I do agree with you; complicated statements should use Ecto's rich and flexible DSL.

github.com/chrismccord/atlas



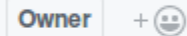
pmontrasio commented Aug 29, 2017



Actually this is 1000 times better than Ecto (much easier to understand and use). Too bad it's dead.



chrismccord commented Aug 29, 2017



I disagree. After I learned more about functional programming and separation of concerns, I changed my view on what a data access library should look like. The changes required to make this library properly designed would have made it look just like Ecto, but worse, and without a nice query DSL :) I challenge you to give Ecto a second pass and really take the time to digest some of the new concepts that may not be familiar from where you're coming from.

Posizionamento

Query semplici	sovradimensionato
Query medie	ok
Query complesse	SQL



</pomodori>

Changeset ed event sourcing

<https://10consulting.com/2017/01/04/building-a-cqrs-web-application-in-elixir-using-phoenix/#example-ecto-repo>

Ricordi meno belli

Ha qualche abitudine strana

Certe volte devi ripetergli le cose

Ci mette un po' a preparare la valigia

Deploy

Travis ok, ma setup non elementare

Deploy

Travis ok, ma setup non elementare

Edeliver ok, ma setup non elementare

Deploy

Travis ok, ma setup non elementare

Edeliver ok, ma setup non elementare

Capisco la tentazione di usare docker



<back to="2015">

Fastidi

alias / import / require all'inizio di ogni file

Mancanza di un framework di autenticazione

Dover chiamare ogni tanto Erlang

Dover scrivere due volte lo schema, nella migrazione e nel modello

Fastidi

alias / import / require all'inizio di ogni file

~~Mancanza di un framework di autenticazione~~

Dover chiamare ogni tanto Erlang

Dover scrivere due volte lo schema, nella migrazione e nel modello

Delizie

Hot reload
iex -S mix
Channels

A highly detailed, golden, circular mechanical structure, possibly a clock face or a complex dial, with intricate patterns and numbers. The structure is composed of concentric rings and radial elements, creating a complex, maze-like appearance. The numbers 1 through 12 are visible, along with various smaller markings and symbols. The overall aesthetic is that of a futuristic or ancient mechanical device. The text '</back>' is overlaid in the center in a white, bold, sans-serif font with a black outline.

</back>

Delizie

Hot reload

ieX -S mix

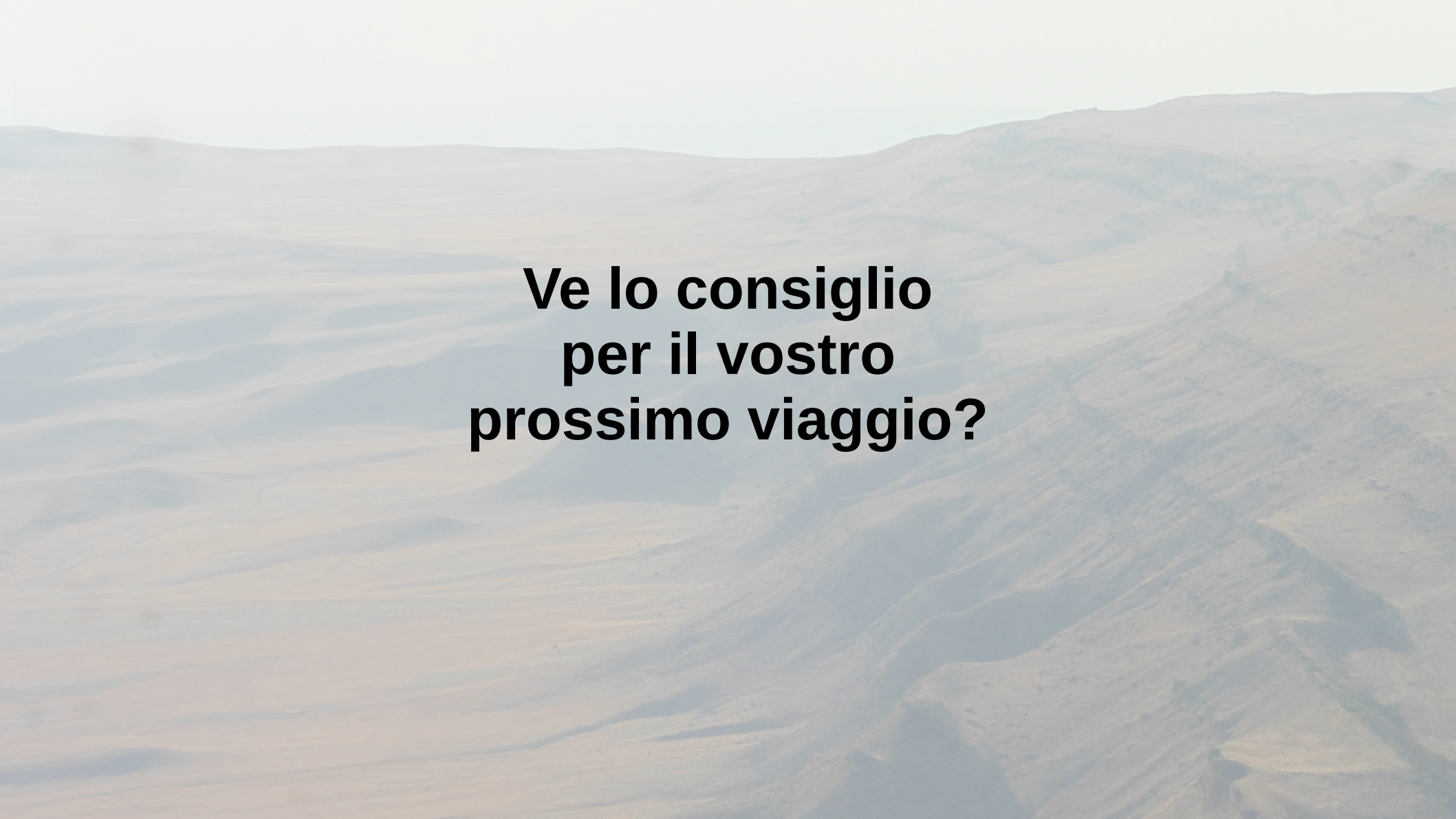
Channels

1 task in background

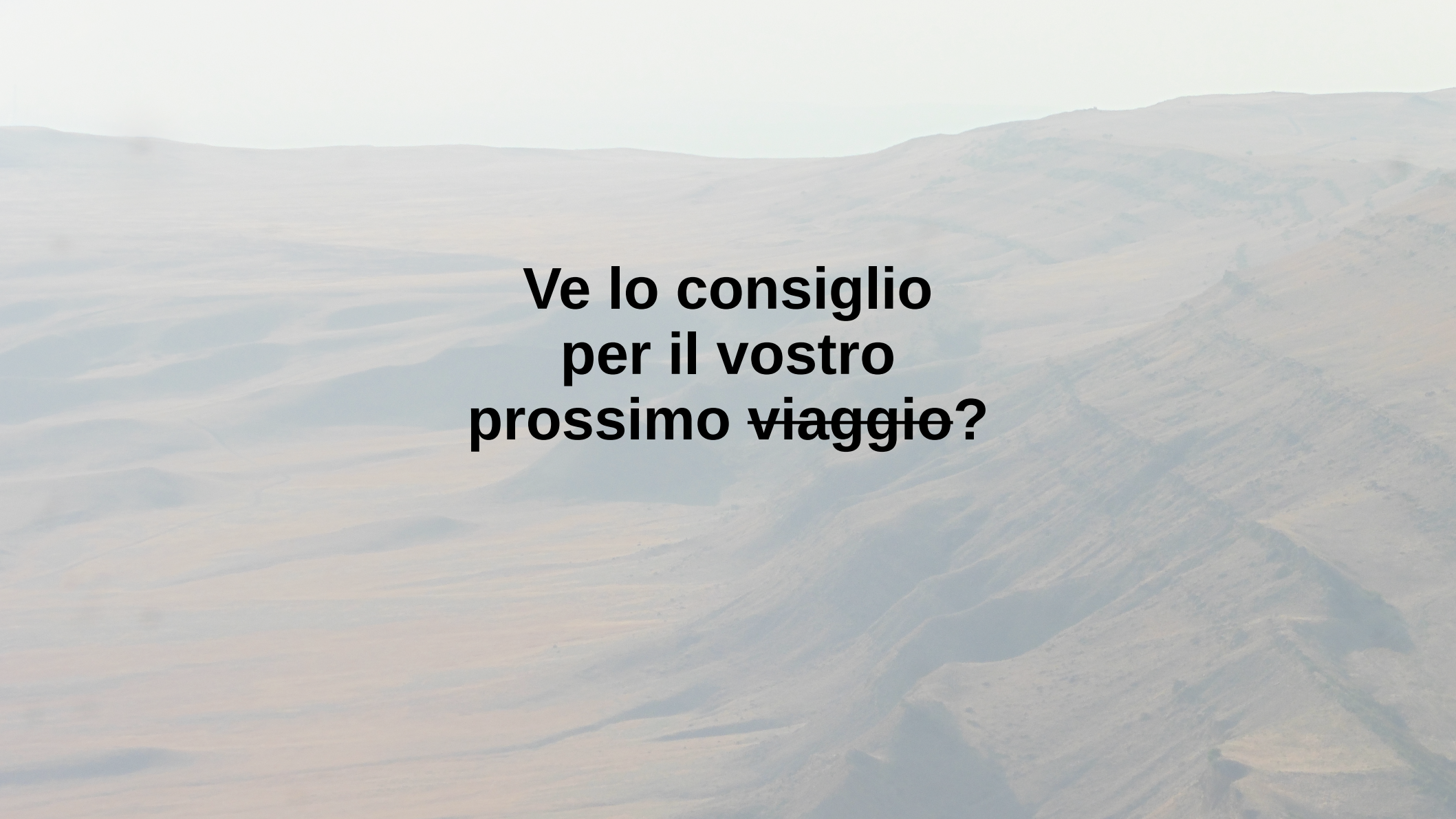
Meno if

Delizia extra: with


```
with {:ok, upload_location} <- get_upload_location(token, document),
    {:ok, gdrive_file_id} <- upload_to_gdrive(token, document,
                                             upload_location),
    {:ok, app_file_id} <- import_into_app(gdrive_file_id)
do
  upload_done()
else
  {:error, reason} -> upload_failed(reason)
end
```

The background of the image is a wide, aerial view of a vast, hazy landscape. The terrain is characterized by rolling hills and deep valleys, all rendered in shades of brown and tan, suggesting a dry or semi-arid environment. The atmosphere is thick with a light, greyish haze, which softens the details of the land and creates a sense of depth and scale. The overall tone is muted and atmospheric.

**Ve lo consiglio
per il vostro
prossimo viaggio?**



**Ve lo consiglio
per il vostro
prossimo ~~viaggio~~?**



**Ve lo consiglio
per il vostro
prossimo progetto?**

Fa quel che deve

Fa quel che deve
Elixir ha più pro che contro

Fa quel che deve
Elixir ha più pro che contro
Se tollerate Ecto...

Fa quel che deve

Elixir ha più pro che contro

Se tollerate Ecto...

Non è peggio di Rails, Django, etc e quindi...

An aerial photograph of a vast, hazy, brownish landscape. The terrain is characterized by rolling hills and deep, winding valleys, creating a complex pattern of light and shadow. The overall color palette is muted, consisting of various shades of tan, beige, and light brown, suggesting a dry or semi-arid environment. The atmosphere is thick with haze, which softens the details of the landscape and creates a sense of depth and scale. The word "Approvato!" is superimposed in the center of the image in a large, bold, black sans-serif font.

Approvato!

A proposito di progetti distribuiti

Integration e deployment automatici ok

Scrivete tutto in un wiki

Tutto il codice visibile a tutti

Domande e contatti

Paolo Montrasio
paolo.montrasio@connettiva.eu

Slide a
connettiva.eu/al-lavoro-con-phoenix