

**Diaspora Installation / Debian / Wheezy official**

■ <https://wiki.diasporafoundation.org/Installation/Debian/Wheezy?db=postgres&mode=development>

\*Database: PostgreSQL

\*Running mode: Development

**Diaspora Installation / Debian / Jessie official**

■ <https://wiki.diasporafoundation.org/Installation/Debian/Jessie?db=postgres&mode=development>

\*Database: PostgreSQL

\*Running mode: Development

**Get the source - Diaspora**

```
$ git clone https://github.com/diaspora/diaspora.git
$ cd diaspora
```

**Configuration**

Copy files

```
$ cp config/database.yml.example config/database.yml
$ cp config/diaspora.yml.example config/diaspora.yml
```

**Bundle**

It's time to install the Ruby libraries required by Diaspora

```
$ gem install bundler
$ bin/bundle install --with postgresql
```

**Database setup**

Double check your config/database.yml looks right and run:

```
$ bin/rake db:create db:schema:load
$ DB=postgres bin/rake db:create db:schema:load // Especificánde la DB.
$ DB=postgres bin/rake db:schema:load // Cuando ya tenemos la db creada.
```

**Start Diaspora**

```
$ ./script/server // It's time to start Diaspora:
$ DB=postgres ./script/server // Especificánde la base de datos, para saltarse algunas notificaciones.
```

**Server diaspora running**

```
$ run -p 3001 // Usando otro puerto, 3000 el predeterminado
```

Instalación de [RedSur?](#) desde los repositorios de cenditel

```
$ git clone https://mayaguaray.cenditel.gob.ve/desarrollo/scm/git/redsursur.git
```

Hay que crear los archivos diaspora.yml y database.yml a partir de los .example que ya vienen por defecto.

En el database.yml modificar el nombre de la base de datos y el usuario:

```
postgres: &postgres
  adapter: postgresql
  host: localhost
  port: 5432
  username: redsur_desarrollo
  password: 123456
  encoding: unicode
```

Además especificar que se va a trabajar con postgresql:

```
common: &common
  # Choose one of the following
  #<<: *mysql
  <<: *postgres
```

y también especificar la db development:

```
development:
  <<: *combined
  database: redsur_desarrollo
```

### Configurar url

En diaspora.yml modificar la URL para que las imágenes que se guardan en la base de datos tengan la dirección especificada.

Pondremos la dirección donde corra nuestro servidor local, ejemplo:

```
url: "http://192.168.12.xxx:3000"
```

### Desactivar el captcha

```
## Enable captcha (default=true)
## Set this to false if you don't want to use captcha for signup process.
enable: false
```

### Otros...

Al momento de hacer el:

```
$ gem install bundler

$ DB=postgres bundle install // Para que no de error con mysql

$ bin/bundle install --with postgresql // Opcional al primero
```

Puede presentar problemas con la instalación del MySQL por lo que se puede comentar esa línea en el Gemfile.

```
ENV['DB'] ||= 'mysql'

#gem 'mysql2', '0.3.18' if ENV['DB'] == 'all' || ENV['DB'] == 'mysql'
gem 'pg', '0.18.1' if ENV['DB'] == 'all' || ENV['DB'] == 'postgres'
```

Especificar la versión de `gem` a usar

```
$ rvm use 2.2 // Para usar el juego de gemas específico para nuestro proyecto, ya que fue la version que instalamos.
$ DB=postgres bin/rake db:migrate RAILS_ENV=development // Creando los esquemas en la db, sobre todo si se agregó un nuevo
$ DB=postgres bundle exec rake db:migrate RAILS_ENV=development // Saltar error de migracion si es necesario, spring, otro
```

### Creando un nuevo modelo

Se agregó un nuevo modelo que corresponde a las Areas de Texto de las Poll: poll\_text\_area.rb

Este modelo tiene una relación uno-a-muchos (has\_many -- belongs\_to) con el modelo Poll.rb (el cual, a su vez, tiene la misma relación con el modelo [StatusMessage?.rb.](#))

### Comando para crear los modelos

```
$ DB=postgres rails generate model PollTextArea text:text
```

Luego modificar el modelo creado.

Generar la migración necesaria para crear asocioaciones entre la instancia creada y su raíz.

```
$ DB=postgres rails g migration AddPollTextAreaToPoll poll:belongs_to
$ DB=postgres rails g migration AddPollToPollTextArea poll:belongs_to
$ DB=postgres bin/rake db:migrate:status // Chequear las migraciones realizadas, cuales se levantaron y cuales estan pendi
$ DB=postgres bin/rake db:migrate RAILS_ENV=development // Para levantar las migraciones creadas, si hay conflictos, se pu
```

Además se modificaron otros modelos.

El controlador status\_messages\_controller.rb recibe de publisher\_view.js (backbone) los parámetros de los controles (p.ej. las areas de texto) a partir de los cuales debe crear las instancias de los modelos (p.ej. el modelo en rails para las areas de texto) que tienen la relación uno-a-muchos (has\_many -- belongs\_to).

Generar la migración necesaria para agregar el area de texto a las tablas correspondientes

```
$ DB=postgres rails g migration AddPollTextAreaToPoll poll:belongs_to
$ DB=postgres rails g migration AddPollToPollTextArea poll:belongs_to
```

### Añadiendo nuevo atributo al profile

```
$ DB=postgres rails g migration AddIsgroupToProfiles
```

Modificar la nueva migracion en /db/migrate --> xxxxxxxxxxxxxx\_add\_isgroup\_to\_profiles.rb

```
class AddIsgroupToProfiles < ActiveRecord::Migration
  def change
    add_column :profiles, :isgroup, :boolean
  end
end
```

```
$ DB=postgres rails g migration AddFrontimageToProfiles
```

```
class AddFrontimageToProfiles < ActiveRecord::Migration
  def change
    add_column :profiles, :front_image, :string
  end
end
```

```
end
```

```
$ DB=postgres bin/rake db:migrate RAILS_ENV=development
```

```
$ DB=postgres bin/rake db:migrate:status // Chequear las migraciones realizadas, cuales se levantaron y cuales estan pendi
```

```
$ bin/rails db:rollback STEP=X // Para moverse entre los estados de las migraciones donde X es el numero a retroceder
```

### Desactivar la confirmacion via mail de las cuentas creadas

(app/models/user.rb) *Editar el fichero*

```
devise :confirmable, :token_authenticatable, :database_authenticatable,
:registerable...
```

Quítale el :confirmable de modo que quede así:

```
devise :token_authenticatable, :database_authenticatable, :registerable...
```

```
devise :confirmable, :token_authenticatable, :database_authenticatable,
:registerable...
```

```
$ gem uninstall <gem-name> // Desinstalar una gema específica, a veces hay conflictos entre versiones de gemas a usar en e
```

### Desactivar aspectos predeterminados de los usuarios

En app/controllers/registrations\_controller.rb comentar linea 16

```
#@user.seed_aspects
```

Borrar usuario de la db: borrar los registros del usuario en la tabla users y en people, queda guardado registro en diaspora\_handle.

```
gpg --keyserver hkp://keys.gnupg.net --recv-keys 409B6B1796C275462A1703113804BB8 Clave publica Michal Papis:
```

### Gema de autenticación

Para el rest se implementará la autenticación con "tokens" mediante la gema (devise\_token\_auth)

In Gemfile

```
# Authentication gem "devise_token_auth"
```

```
postrequest.py
```

```
#!/ coding: utf-8
```

```
import httplib, urllib
```

```
import json
```

```
params = json.dumps("utf8": "✓", "authenticity_token": "", "user": {"email": "dieciocho@quince.org", "username": "dieciocho", "p
```

```
headers = {"Content-Type": "application/json", "Accept": "*/*"}
```

```
host = "192.168.12.156"
```

```
port = 3000
```

```
conn = httplib.HTTPConnection(host, port)
```

```
conn.request("POST", "/users", params, headers)
```

```
response = conn.getresponse()
```

```
print "Response: "
```

```
print response.status, response.reason,
```

```
print "Body: "
```

```
print response.read()

conn.close()
```

### Agregando el nuevo campo phone al profile del user

```
$ DB=postgres rails g migration AddPhoneToProfiles
```

Editamos el diaspora/db/migrate/20161121193410\_add\_phone\_to\_profiles.rb

```
class AddPhoneToProfiles < ActiveRecord::Migration
  def change
    add_column :profiles, :phone, :string
  end
end
```

Luego migramos...

```
$ DB=postgres bin/rake db:migrate RAILS_ENV=development
```

Chequeamos las migraciones...

```
$ DB=postgres bin/rake db:migrate:status
```

Si realizó correctamente la migración nos mostrará...

```
.
.
.
up    20160307142216  Cleanup handles
up    20160327103605  Add author id index to participations
up    20161121193410  Add phone to profiles
```

Y en la tabla Profiles de la db debe aparecer el campo:

```
phone | character varying
```