

# Laboratorio de Computación IV

## Clase 24

*Andrés Fortier*

# Repaso



- Benchmarks
  - Importancia en saber qué medir y cómo medirlo.
  - N corridas (descartar la primera).
  - Garbage collector.
  - Considerar la distribución.
- Árboles binarios de búsqueda.

# Repaso



- Indices de BD para mejorar tiempos de búsquedas
  - Costo de inserciones y eliminaciones para mantener el árbol balanceado.
  - Costo de espacio en disco.
- Pre-cargar modelos relacionados para reducir la cantidad de queries.

# Ubicación Geográfica



- La ubicación en un mapa se suele dar por un par de coordenadas (latitud, longitud).
- Las distancias deben medirse sobre la superficie de la esfera.
- Comencemos representando la ubicación de una persona
  - Parte de su perfil. Requiere una migración.
  - Visualización usando google maps.

# Ubicación Geográfica

- Extendamos el modelo del perfil

```
$ bin/rake db:reset  
$ bin/rails g migration AddLatLonToProfile lat:decimal  
lon:decimal  
$ bin/rake db:migrate
```

- Agreguemos una gema para el manejo de mapas

```
/Gemfile  
  
..  
gem 'gmaps4rails', '~> 2.1.2'  
...
```

```
$ bundle install
```

# Ubicación Geográfica

- Configuremos el uso de las librerías de javascript

```
/app/assets/javascripts/application.js
```

```
...  
//= require gmaps/google  
//= require_tree .
```

# Ubicación Geográfica

- Agreguemos las librerías necesarias en el html

```
/app/views/layouts/application.html.erb
```

```
...  
<script  
src="//cdnjs.cloudflare.com/ajax/libs/underscore.js/1.8.3/  
underscore-min.js" type="text/javascript"></script>  
<script src="//maps.google.com/maps/api/js?  
v=3.13&sensor=false&libraries=geometry"  
type="text/javascript"></script>  
<script src='//google-maps-utility-library-  
v3.googlecode.com/svn/tags/markerclustererplus/2.0.14/src/  
markerclusterer_packed.js'  
type='text/javascript'></script>  
<%= stylesheet_link_tag 'application', media: 'all',  
'data-turbolinks-track' => true %>  
...
```

# Ubicación Geográfica

```
/app/controllers/profiles_controller.rb
```

```
def profile_params
  params
    .require(:profile)
    .permit(
      :first_name,
      :last_name,
      :date_of_birth,
      :lat,
      :lon,
      phones_attributes: [
        :id,
        :phone_type,
        :number,
        :_destroy
      ]
    )
end
```



# Ubicación Geográfica

- Modifiquemos la vista para que
  - Se muestre en dos columnas.
  - Llame a una función javascript para mostrar el mapa.

```
/app/views/profiles/edit.html.erb
```

```
<h1>Editing profile</h1>
```

```
<div class="row">
```

```
  <div class="col-md-6">
```

```
    <%= form_for @profile do |f| %>
```

```
    ...
```

# Ubicación Geográfica

```
/app/views/profiles/edit.html.erb
```

```
</div>
<div class="col-md-6">
  <div id="profile_map" style="width: 550px; height:
350px;">
    </div>
  </div>
</div>

<%= javascript_tag do %>
  $(document).ready(function() {
    var profile = {lat: -40.1552557, lon: -71.3472031};
    show_map_for(profile);
  });
<% end %>
```

# Ubicación Geográfica

```
/app/assets/javascripts/google_maps.js

function show_map_for(profile) {
  if ((profile.lat == null) || (profile.lon == null) ) {
    return;
  }
  handler = Gmaps.build('Google');
  handler.buildMap({ provider: {}, internal: {id:
'profile_map'}}}, function(){
    var markers = handler.addMarkers([
      {
        "lat": profile.lat,
        "lng": profile.lon,
      }
    ]);
    handler.bounds.extendWith(markers);
    handler.fitMapToBounds();
    handler.getMap().setZoom(14);
  });
}
```

# Ubicación Geográfica

```
/app/models/user.rb
```

```
...
  accepts_nested_attributes_for :profile, :allow_destroy
=> true

  after_create :create_profile_if_missing

  def create_profile_if_missing
    if !profile.present?
      Profile.create(user: self)
    end
  end
end
...
```

# Ubicación Geográfica

```
/db/seeds.rb
```

```
joeUser = User.new(  
  :email              => "joe@example.com",  
  :password            => "12345678",  
  :password_confirmation => "12345678",  
  :profile_attributes => {  
    :first_name      => "Joe",  
    :last_name       => "Doe",  
    :date_of_birth   => "1980-06-22".to_date,  
    :lat             => -40.1552557,  
    :lon             => -71.3472031  
  }  
)
```

```
$ bin/rake db:reset
```

# Ubicación Geográfica

- Modifiquemos la vista

```
/app/views/profiles/edit.html.erb
```

```
...  
<%= javascript_tag do %>  
  $(document).ready(function() {  
    var profile = <%= @profile.to_json.html_safe %>;  
    show_map_for(profile);  
  });  
<% end %>
```

# Ubicación Geográfica



- Editar la ubicación
  - Tener dos campos ocultos lat y lon.
  - Mostrar un mapa en blanco si no hay datos de ubicación.
  - Cambiar la ubicación al clickear en el mapa.

# Ubicación Geográfica

- Modificaciones en la vista

```
/app/views/profiles/edit.html.erb
```

```
<h1>Editing profile</h1>
```

```
<%= form_for @profile do |f| %>
```

```
  <div class="row">
```

```
    ...
```

```
    <div class="form-group">
```

```
      <%= f.label :date_of_birth %>
```

```
      <%= f.date_field :date_of_birth ... %>
```

```
    </div>
```

```
    <%= f.hidden_field :lat, {id: :profile_lat} %>
```

```
    <%= f.hidden_field :lon, {id: :profile_lon} %>
```

```
  ...
```



# Ubicación Geográfica

- Modificaciones en Javascript

```
/app/assets/javascripts/google_maps.js
```

```
function show_map_for(profile) {  
  var markerOnMap;  
  var handler = Gmaps.build('Google');  
  handler.buildMap({ provider: {}, internal: {id:  
'profile_map'}}}, function(){  
  
    if ((profile.lat !== null) && (profile.lon !== null))  
    {  
      markerOnMap = handler.addMarker(  
        {  
          "lat": profile.lat,  
          "lng": profile.lon,  
        }  
      );  
    }  
  }  
}
```

# Ubicación Geográfica

```
/app/assets/javascripts/google_maps.js
```

```
...
    handler.bounds.extendWith([markerOnMap]);
    handler.fitMapToBounds();
    handler.getMap().setZoom(15);
} else {
    handler.fitMapToBounds();
    handler.map.centerOn([-40.1552557, -71.3472031]);
    handler.getMap().setZoom(14);
}
});
```

# Ubicación Geográfica

```
/app/assets/javascripts/google_maps.js
```

```
function placeMarker(location) {  
  if (markerOnMap) {  
    markerOnMap.serviceObject.setPosition(location);  
  }  
  else {  
    markerOnMap = new google.maps.Marker({  
      position: location,  
      map: handler.getMap()  
    });  
  }  
}
```

# Ubicación Geográfica

```
/app/assets/javascripts/google_maps.js
```

```
  google.maps.event.addListener(handler.getMap(), 'click',  
function(event) {  
  placeMarker(event.latLng);  
  $("#profile_lat").val(event.latLng.lat());  
  $("#profile_lon").val(event.latLng.lng());  
});  
}
```

# Ubicación Geográfica - Búsquedas



- Tipos de búsquedas
  - Dentro de un polígono arbitrario.
  - Dentro de un rectángulo (bounding box).
  - Por distancia.

# Ubicación Geográfica

- Agreguemos una gema para hacer queries sobre distancias

```
/Gemfile
```

```
..  
gem 'geocoder', '~> 1.2.9'  
...
```

```
$ bundle install
```

# Ubicación Geográfica

- Modifiquemos el perfil

```
/app/models/profile.rb
```

```
class Profile < ActiveRecord::Base
  belongs_to :user
  has_many :phones, :dependent => :destroy
  accepts_nested_attributes_for :phones, :allow_destroy =>
true
  geocoded_by :address, :latitude => :lat, :longitude
=> :lon
end
```

# Ubicación Geográfica

- Extendamos nuestro generador

```
/app/services/benchmark_seeds.rb
```

```
class BenchmarkSeeds
```

```
...
```

```
def seed_profiles
```

```
  generator = Random.new
```

```
  ActiveRecord::Base.transaction do
```

```
    User.destroy_all
```

```
  end
```

```
  10.times do |i|
```

```
    ActiveRecord::Base.transaction do
```



# Ubicación Geográfica

```
/app/services/benchmark_seeds.rb
```

```
1.upto(5000) do |j|
  id = i * 5000 + j
  lat = generator.rand(-40.1638078..-40.1508152)
  lon = generator.rand(-71.3555931..-71.3399183)
  user = User.create!(
    :email          => "user#{id}@example.com",
    :password        => "12345678",
    :password_confirmation => "12345678",
    :profile_attributes => {
      :first_name    => "User#{id}",
      :last_name     => "Doe",
      :date_of_birth => "1980-06-22".to_date,
      :lat           => lat,
      :lon           => lon
    }
  )
end
end
```

# Ubicación Geográfica



- En una consola

```
$ bin/rails console  
> Profile.near([-40.158, -71.345], 0.4, :units =>  
:km).all.length
```

# Ubicación Geográfica

- Volvamos al perfil

```
/app/models/profile.rb
```

```
class Profile < ActiveRecord::Base
  ...
  geocoded_by :address, :latitude => :lat, :longitude
=> :lon
  reverse_geocoded_by :lat, :lon

  def address=(v)
    p v
  end

  def address
    "Perito Moreno 1500, San Martin de los Andes, Neuquén,
Argentina"
  end
end
```

# Ubicación Geográfica

- En una consola

```
$ bin/rails console
> Profile.first.reverse_geocode
Profile Load (0.2ms)  SELECT  "profiles".* FROM "profiles"
ORDER BY "profiles"."id" ASC LIMIT 1
Geocoder: HTTP request being made for
http://maps.googleapis.com/maps/api/geocode/json?
language=en&latlng=-40.15837415131922%2C-
71.35559010123636&sensor=false
"Avenida San Martín 402-500, San Martin de los Andes,
Neuquén, Argentina"
=> "Avenida San Martín 402-500, San Martin de los Andes,
Neuquén, Argentina"
```

# Ubicación Geográfica

- En una consola

```
$ bin/rails console
> Profile.first.geocode
  Profile Load (0.3ms)  SELECT  "profiles".* FROM "profiles"
  ORDER BY "profiles"."id" ASC LIMIT 1
Geocoder: HTTP request being made for
http://maps.googleapis.com/maps/api/geocode/json?
address=Perito+Moreno+1500%2C+San+Martin+de+los+Andes
%2C+Neuqu%C3%A9n%2C+Argentina&language=en&sensor=false
=> [-40.1524717, -71.3448425]
```

# Geocoding / reverse geocoding

- Se suele colocar como hooks en `after\_validation`
- Hacerlo automático y bidireccional tiene ciertos casos a considerar.
- Evitar requerimientos innecesarios.

```
/app/models/profile.rb
```

```
after_validation :geocode, if: ->(obj)  
{ obj.address.present? and obj.address_changed? }
```