



Comparte tus trayectos diarios



Motivación

Tecnologías

Proceso

Conclusiones

Ernesto Gimeno Sánchez
<https://dailyjourney.netlify.app/home>



<https://github.com/EGimenoS/dailyjourney>
<https://github.com/EGimenoS/dailyjourneyapi>

¿Por qué DailyJourney?



¿Qué es?

¿Qué aporta como producto?

¿Qué me aporta a mí?

Qué es DailyJourney

DailyJourney es una aplicación web que pone en contacto a personas que realizan trayectos recurrentes a destinos cercanos

2,5M de viajes diarios por motivo de trabajo en Barcelona (EMEF12 2015)

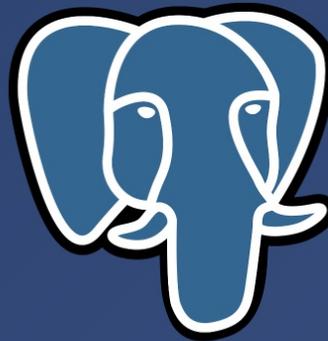
¿Qué aporta como producto?



¿Qué me aporta a mí?

- Realizar una aplicación completa fullstack
- Experimentar con servicios third party
- Ir más allá del CRUD
- Explorar los diferentes aspectos de un producto tecnológico

Tecnologías



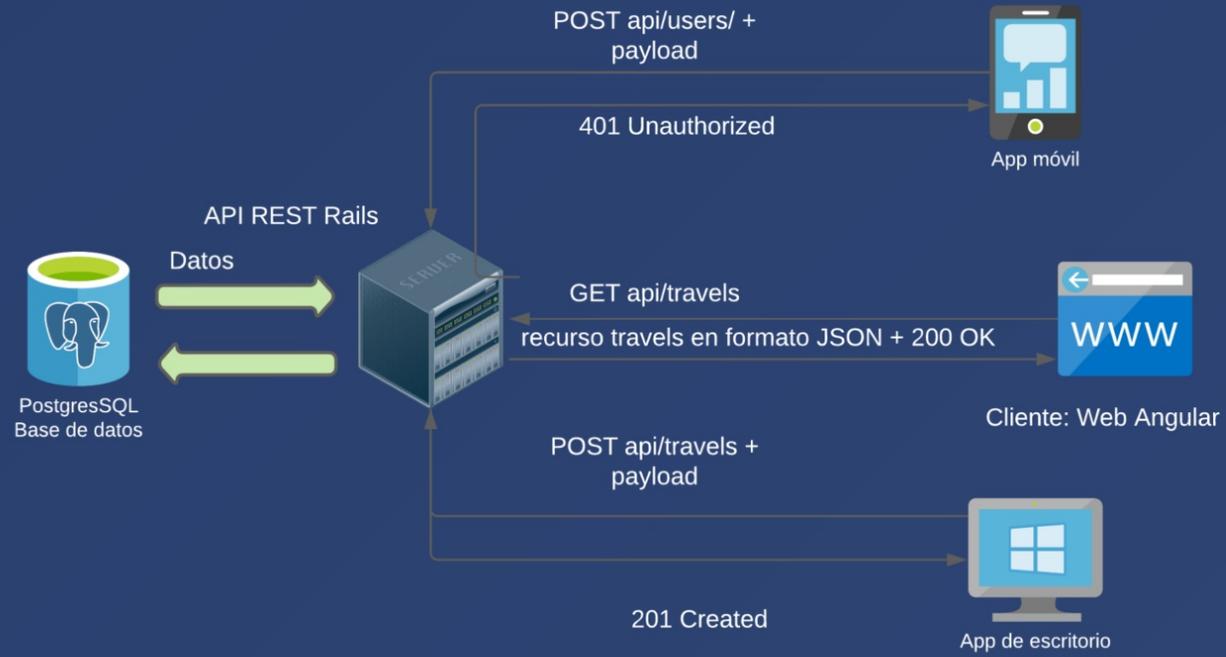
Arquitectura

Rails

Angular

Otras
tecnologías

Arquitectura de la aplicación



Ruby On Rails



Framework MVC programado en Ruby



Convention over configuration



ActiveRecord



Modo solo API



Gran comunidad open source

Angular. El framework de Google para crear SPA's

Framework completo

- Claras convenciones
- Soluciones dentro del framework a problemas habituales
- Gran comunidad y recursos

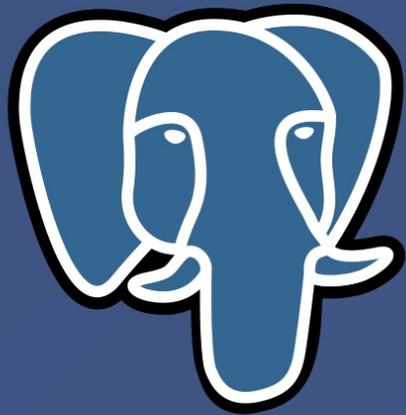
TypeScript

- Superset tipado de Javascript
- Mejora experiencia de desarrollo (intellisense)
- Detección de errores en tiempo de desarrollo

Multiplataforma

- PWA's
- Angular Universal
- Ionic, NativeScript...

Otras tecnologías utilizadas





Proceso

1. Toma de decisiones inicial y planificación
2. Setup de proyectos e implementación de autenticación
3. Desarrollo de features
4. Pruebas, retoques y finalización de documentación

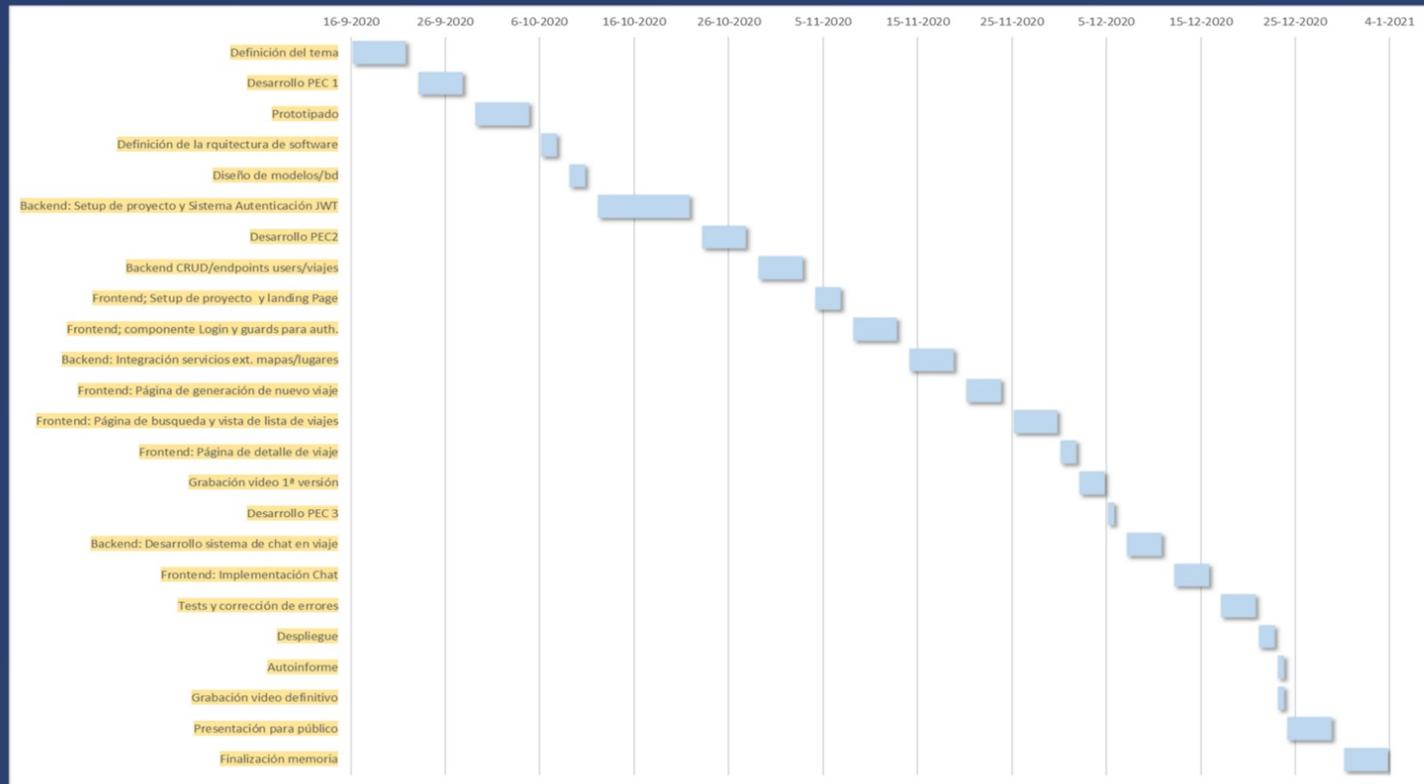
Planificación

**Autenticación/
autorización**

**Integración
con servicio
de mapas**



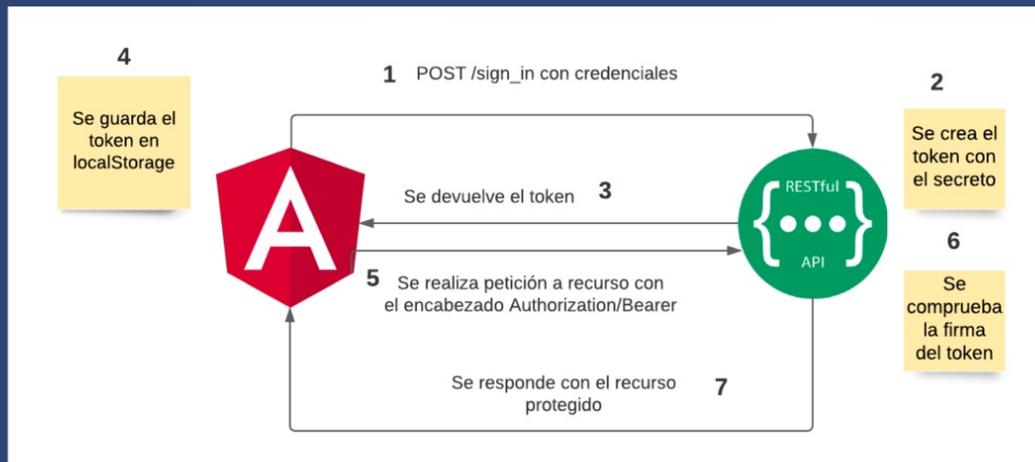
Diagrama de Gantt



Autenticación/Autorización

Rails no dispone de una convención para la autenticación de API's...

Investigación previa: OAuth2, Firebase, Auth0...y finalmente JWT



✓	Flujo de registro/login
	Protección de todos los endpoints
	Cambio de password
✗	Flujo refresh token
	Flujo confirmar email
	Flujo recordar contraseña

Integración con servicio de mapas

- Necesario para sugerir al usuario direcciones válidas
- Se estudian alternativas más populares, como Google Places API
- Finalmente se usa Here Maps en su endpoint Discover: Fácil de usar y potente tier Freemium.

Cambiar de este servicio a otro distinto es fácil:
patrón Adapter

```
module Adapter
  class HereMaps
    API_KEY = ENV['here_maps_api_key']

    def initialize(base_url, query, at = '39.48728,-0.36593')
      @base_url = base_url
      @config = set_config
      @query = query
      @at = at
    end

    def autosuggest_addresses
      req_params = { at: @at, q: @query }
      results = RestClient.get @base_url, { params: @config.merge(req_params) }
      results = JSON.parse(results)
      results['items'].map do |item|
        { address: item['title'], latitude: item['position']['lat'], longitude: item['position']['lng'] }
      end
    end

    private

    def set_config
      { apikey: API_KEY, limit: 5 }
    end
  end
end
```

Proyección a futuro y conclusiones

```
<defs>  
  <linearGradient x1="100%" y1="0%" x2="0%" y2="0%">  
    <stop stop-color="black" offset="0%"/>  
    <stop stop-color="black" offset="100%"/>  
  </linearGradient>  
</defs>  
<rect width="600" height="450" rx="0" fill="url(#media-control)"/>  
</svg>  
<div class="media-control">  
  <svg width="96" height="96" viewBox="0 0 96 96">  
    <defs>  
      <linearGradient x1="0%" y1="0%" x2="0%" y2="100%">  
        <stop stop-color="white" offset="0%"/>  
        <stop stop-color="white" offset="100%"/>  
      </linearGradient>  
      <filter x="50%" y="50%" width="100%" height="100%">  
        <feOffset dy="15" in="Source" result="offsetBlur"></feOffset>  
        <feGaussianBlur stdDeviation="10" in="offsetBlur" result="blur"></feGaussianBlur>  
        <feColorMatrix values="0 0 0 0 0" type="matrix" in="blur" result="colorMatrix"></feColorMatrix>  
      </filter>  
      <fill-rule="evenodd"></fill-rule>  
    </defs>  
    <rect width="96" height="96" rx="48" fill="url(#media-control)"/>  
    <rect width="96" height="96" rx="48" fill="url(#media-control)"/>  
  </svg>  
</div>
```

Mejoras de calidad de código

Mejoras tecnológicas

Nuevas Features

Retrospectiva

Mejoras de escalabilidad/mantenibilidad

Suite de tests

Refactorizar componentes

Incluir sistema CI/CD

↓

Menos bugs y mejor experiencia de desarrollo

Mejoras de tipo tecnológico/ de plataforma

1

Uso de websockets

2

Implementar Oauth2

3

Convertir a PWA

4

Explorar Angular Universal

Nuevas features

- 1 Registro con proveedores sociales
- 2 Notificaciones in-app e email
- 3 Subida de avatar propio
- 4 Añadir verificación de conductores
- 5 Añadir calendario de viajes
- 6 Permitir definir radio de búsqueda

Retrospectiva

Objetivos principales

- ✓ Desarrollo app completa
- ✓ Desarrollo SPA
- ✓ Desarrollo API REST
- ✓ Desarrollo responsive

Objetivos secundarios

- ✓ Desplegar en producción
- ✓ Conectarse a servicios externos
- ✓ Uso de github, linters, herramientas modernas
- ✓ Documentar adecuadamente el proceso
- ✗ Implementar tests unitarios y e2e

Y la demo...



<https://dailyjourney.netlify.app>

¡Muchas gracias por su atención!



Comparte tus trayectos diarios



Motivación

Tecnologías

Proceso

Conclusiones

Ernesto Gimeno Sánchez
<https://dailyjourney.netlify.app/home>



<https://github.com/EGimenoS/dailyjourney>
<https://github.com/EGimenoS/dailyjourneyapi>