Universitat Oberta de Catalunya

Trabajo Fin de Grado

Diseño e implementación de una base de datos de eventos

Autor:

Daniel Bertrand Pilot Combarro

Directores:

Ismael Campanario Cabrera

Josep Curto Díaz

Introducción

1.1 Contexto y justificación del Trabajo

Contexto

- Fin de la pandemia de COVID-19.
- Recuperación de eventos presenciales: niveles de asistencia previos a la pandemia.
- Dificultad para descubrir eventos de interés locales.
- Oportunidades de negocio: información y venta de entradas.
- Avances tecnológicos: geolocalización y personalización.

1.2 Objetivos del Trabajo

Objetivo Principal

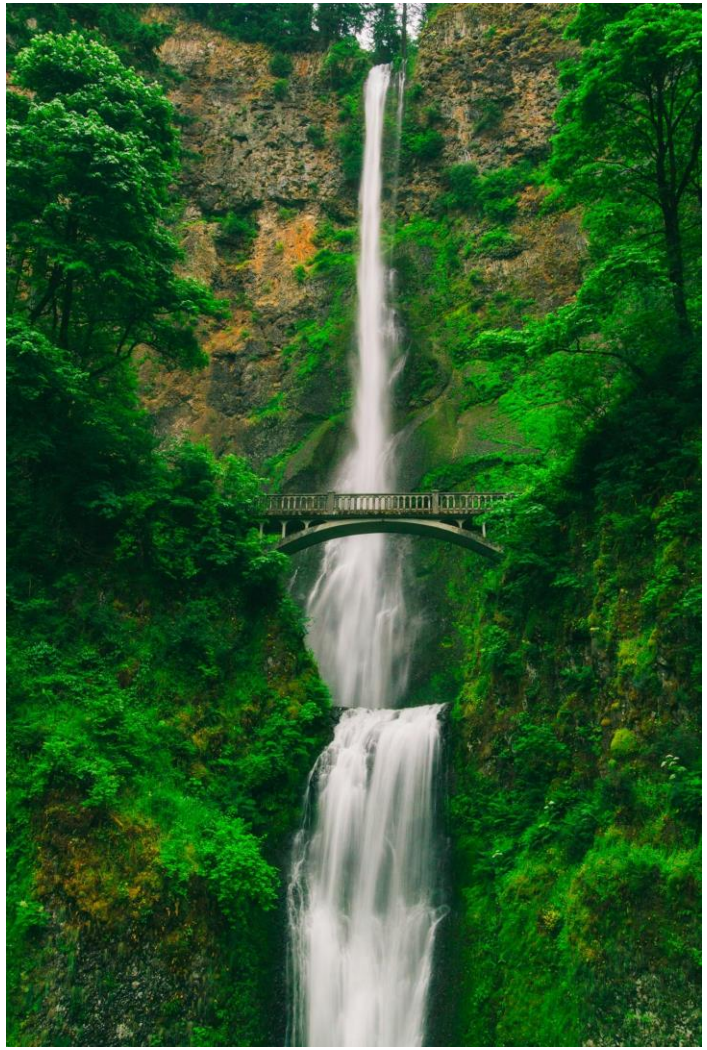
- Diseñar e implementar una **base de datos de eventos**.
- Uso por un sistema de **gestión de eventos**.
- Permitir la **venta de entradas**.

1.2 Objetivos del Trabajo

Objetivos específicos

- Estructura eficiente.
- Consultas optimizadas.
- Trazabilidad de acciones.
- Datos estadísticos.
- Procedimientos alta, baja, modificación.
- Seguridad de los datos
- Mostrar conocimientos adquiridos en el grado.

1.3 Enfoque y método seguido



Modelo en cascada

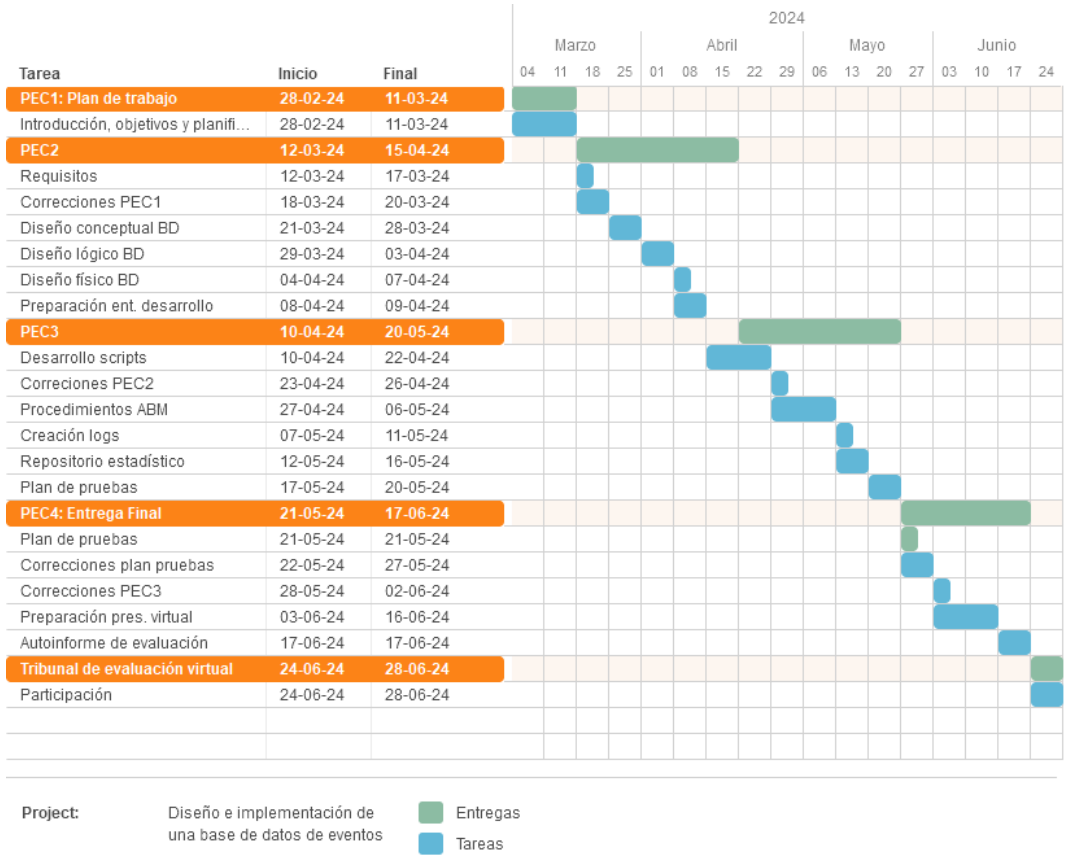
- Modelo lineal.
- Fases marcadas inicialmente.
- Hitos con fechas específicas.
- Dependencia entre fases.
- Sencillez de implementación.

Fases:

1. Requisitos.
2. Diseño.
3. Implementación.
4. Pruebas.
5. Entregable

Imagen: Pixabay. Pexels

1.4 Planificación del Trabajo



Requisitos del Producto

2.1 Requisitos funcionales

Requisitos funcionales

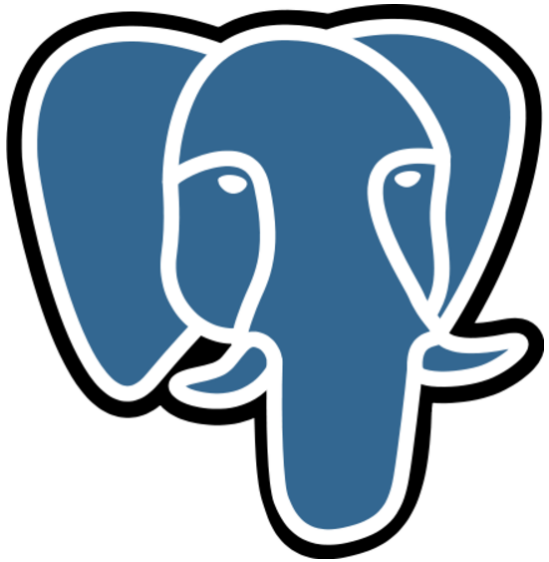
- Se han obtenido **21** requisitos funcionales.
- Base de datos con **información de eventos** y su **ubicación**.
- Implementar un sistema de **usuarios**.
- Implementar un sistema de **venta de entradas**.
- Repositorio **estadístico**.

2.2 Requisitos no funcionales

Requisitos no funcionales

- Se han obtenido **13** requisitos no funcionales.
- Base de datos **relacional**.
- Diseño **escalable**.
- **Procedimientos ABM** en entidades relevantes.
- **Diagrama UML** para diseño conceptual.
- **Trazabilidad** de cambios.
- Sistema de **pruebas**.

2.3 Elección del SGBD: PosgreSQL



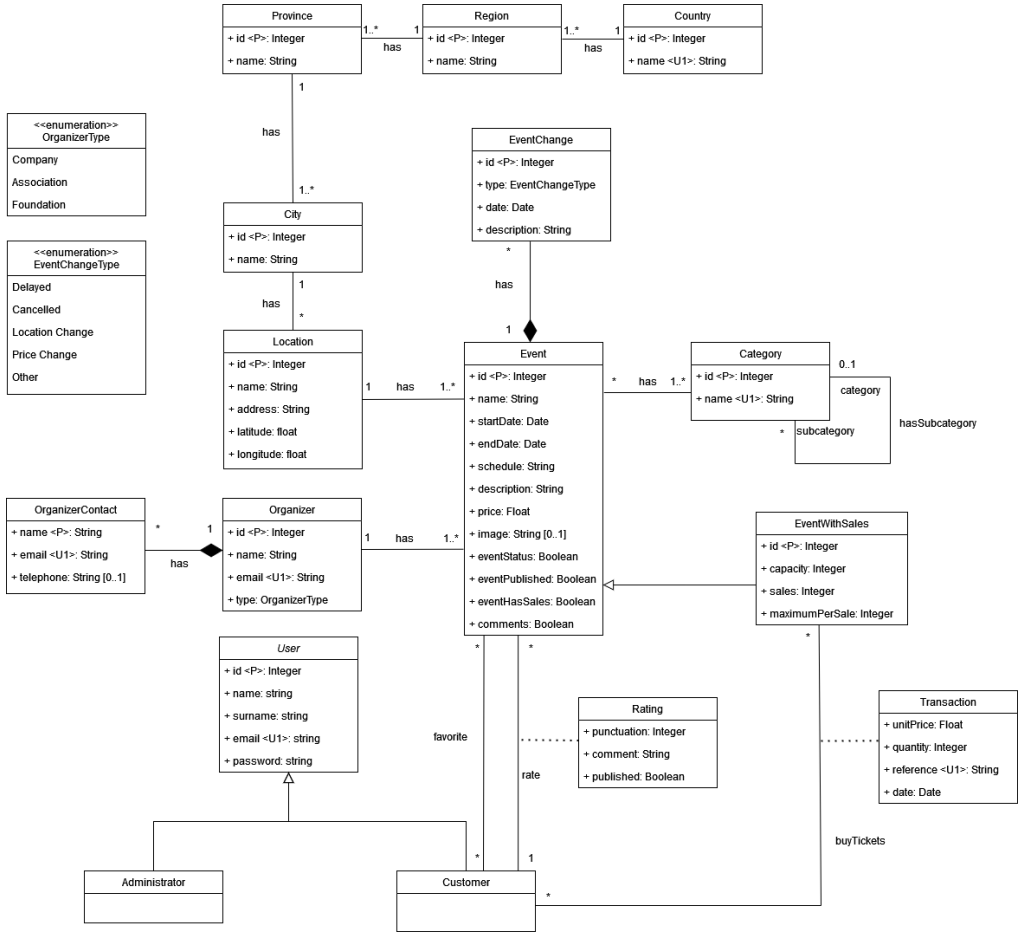
¿Por qué PostgreSQL?

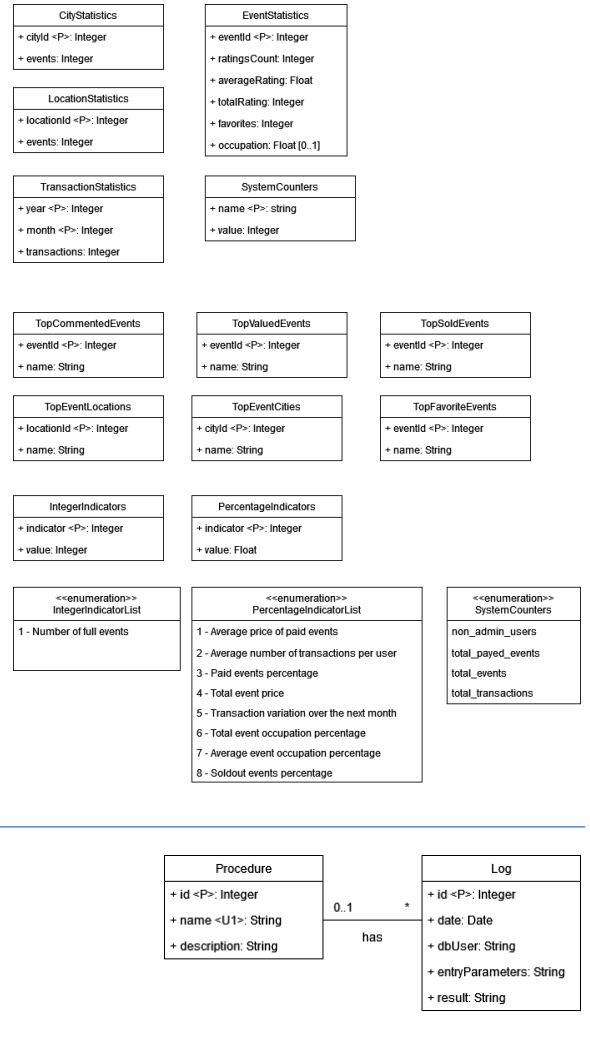
- Código abierto y amplia comunidad.
- Objeto-relacional.
- Soporte multiplataforma.
- Extensible.
- Soporte ACID.
- Procedimientos almacenados y disparadores.
- Rendimiento.
- Soporte Docker.
- Interés personal.

Imagen: PostgreSQL

Diseño

3.1 Diseño conceptual





3.2 Diseño lógico

Country (id, **name**)

Region (id, **name**, **country_id**)
{country_id} is foreign key to Country

Province (id, **name**, **region_id**)
{region_id} is foreign key to Region

City (id, **name**, **province_id**)
{province_id} is foreign key to Province

Location (id, **name**, **city_id**, **address**, **latitude**, **longitude**)
{city_id} is foreign key to City

Category (id, **name**, parent_category)
{parent_category} is foreign key to Category

Organizer (id, **name**, **email**, **type**)

OrganizerContact (name, organizer_id, **email**, telephone)
{organizer_id} is foreign key to Organizer

Event (id, **name**, **start_date**, **end_date**, **schedule**, **description**, **price**, image, **event_status**, **event_published**, **event_has_sales**, **comments**, **organizer_id**, **location_id**)
{organizer_id} is foreign key to Organizer
{location_id} is foreign key to Location

3.2 Diseño físico

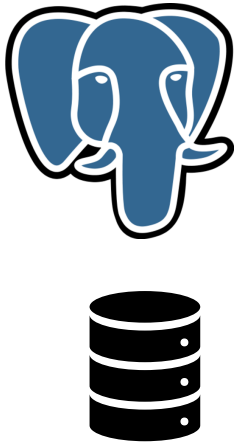
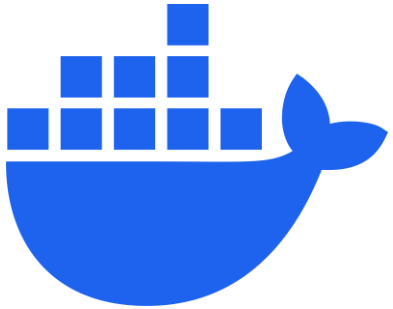


Imagen: PostgreSQL

Implementación

4.1 Configuración del entorno de desarrollo

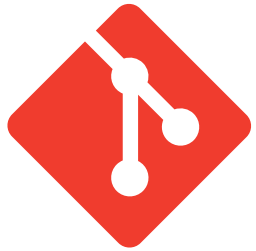


Docker

- Plataforma de código abierto para ejecutar **contenedores**.
- Usaremos un contenedor que ejecute **PostgreSQL**.
- Instalaremos y configuraremos las **dependencias** de nuestro contenedor.

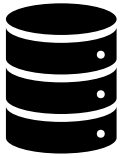
GIT

- Software de **control de versiones**.
- Mantendremos un **repositorio** con los cambios realizados en el proyecto.

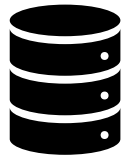


Imágenes: Docker y Git

4.2 Espacio virtual

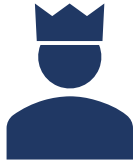


Espacio operacional



Espacio estadístico

4.3 Usuarios



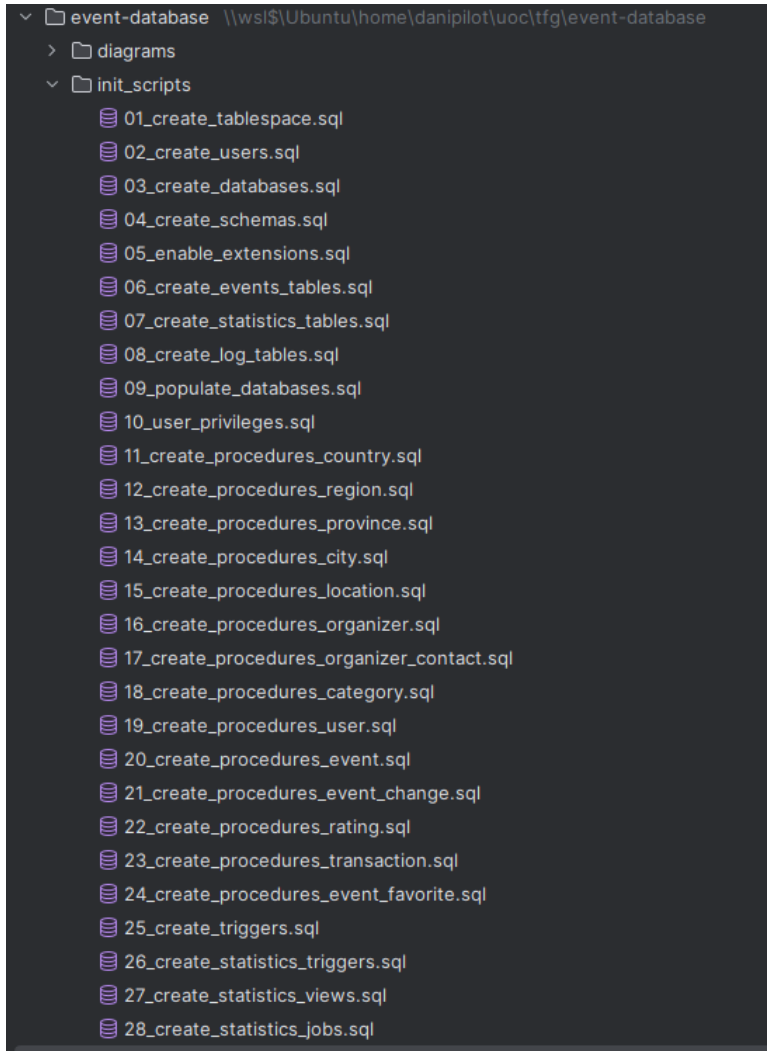
postgres — Super administrador

event_user — Administrador eventos

log_user — Administrador registros

statistics_user — Administrador estadísticas

4.4 Instalación e inicialización de la base de datos

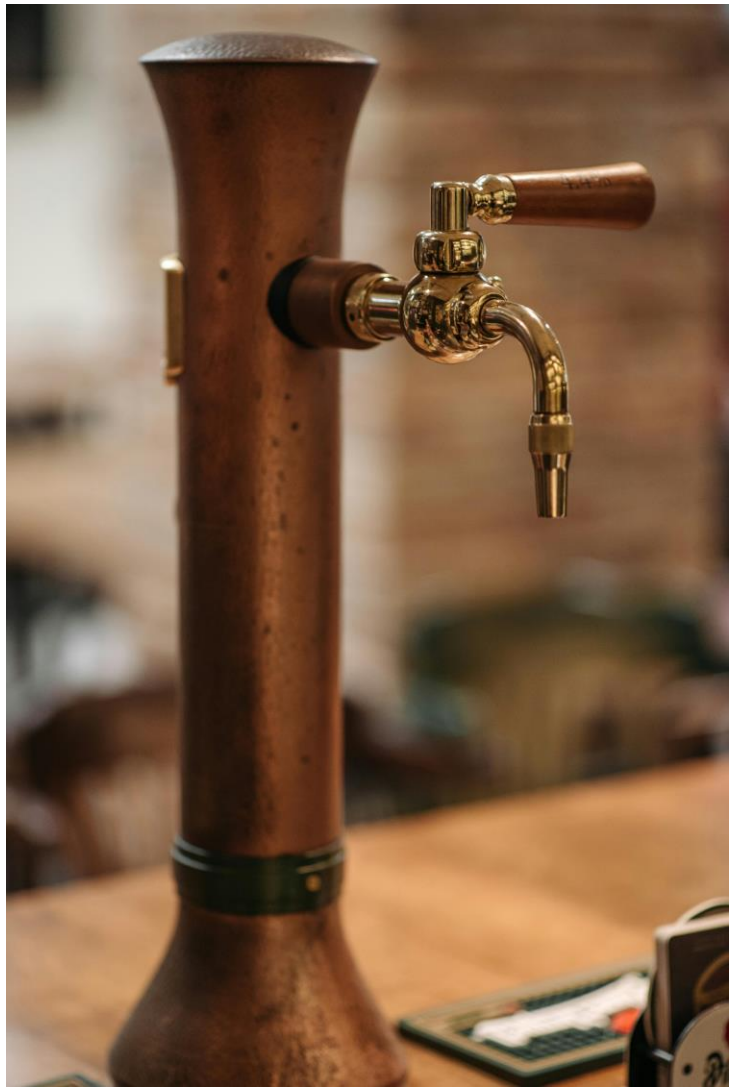


- Se ejecutan los scripts SQL de forma **secuencial**, en orden alfabético.
- Se activan las **extensiones**, se crean los **usuarios**, las **tablas** y **relaciones**, se **pobla** la base de datos, se crean los **procedimientos**, las **vistas** etc.

Eventos programados con **pg_cron** para el repositorio estadístico.

Imagen: Citusdata

4.5 Pruebas



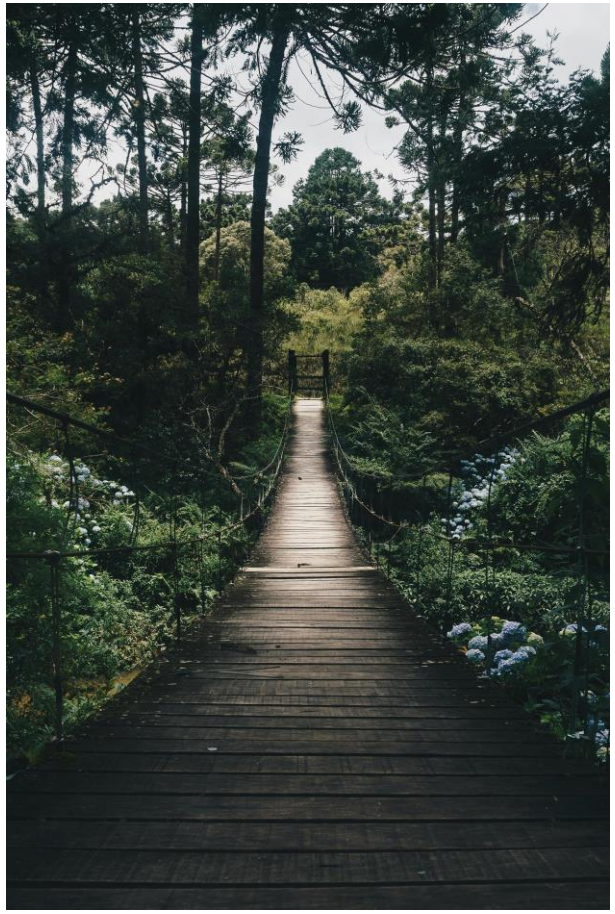
- Trabajamos con la extensión **pgTAP**.
- Implementamos **pruebas unitarias** del desarrollo realizado.
- Las pruebas nos permiten:
  - **Validar** que la funcionalidad desarrollada cumple con los **requisitos**.
  - Hacer **pruebas de regresión**.

```
/app/tests/47_statistics_jobs.sql ..........
1..12
ok 1 - Top commented events updated
ok 2 - Top commented events updated in correct order
ok 3 - Top valued events updated
ok 4 - Top valued events updated in correct order
ok 5 - Top sold events updated
ok 6 - Top sold events updated in correct order
ok 7 - Top event locations updated
ok 8 - Top event locations updated in correct order
ok 9 - Top event cities updated
ok 10 - Top event cities updated in correct order
ok 11 - Top favorite events updated
ok 12 - Top favorite events updated in correct order
ok
All tests successful.
Files=47, Tests=610,  3 wallclock secs ( 0.28 usr  0.06 sys +  1.43 cusr  0.28 csys =  2.05 CPU)
Result: PASS
root@cde459548df5:/#
```

Imagen: Pavel Danilyuk. Pexels

Conclusiones

5. Conclusiones

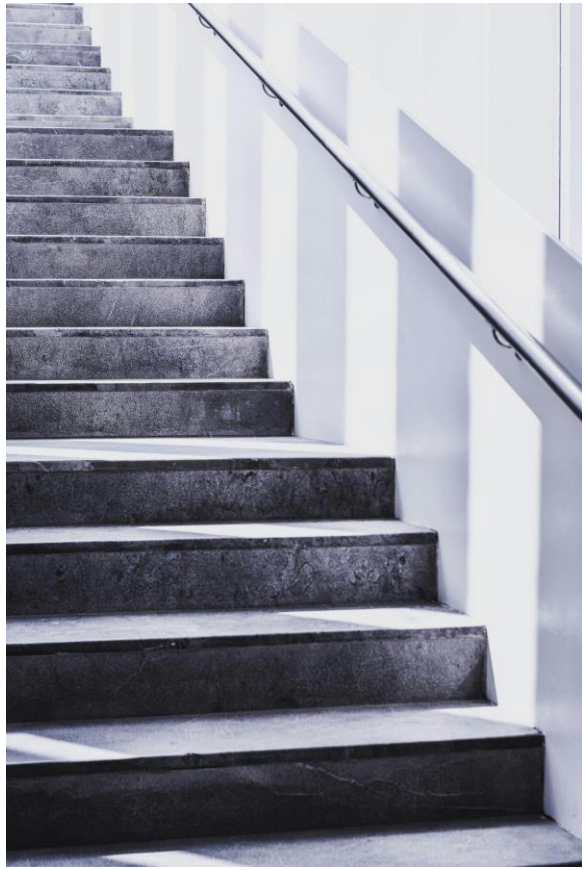


- Se han **cumplido los objetivos** del proyecto.
- Algunas de las estimaciones realizadas eran **demasiado optimistas**.
- Hemos elegido el **método correcto**.
- Se han podido **mitigar** los errores con una mayor **dedicación**.
- Es recomendable realizar las **pruebas unitarias** durante el desarrollo de las funcionalidades.
- Se han podido poner en práctica los **conocimientos** y **habilidades transversales** adquiridos durante el grado.

Imagen: Kaique Rocha. Pexels

Trabajo futuro

6. Trabajo futuro



- Permitir **más de una transacción** por evento y usuario.
- Mejora del sistema de **ventas de entradas**.
- Almacenar más **información del usuario**.
- Implementar **códigos de descuento**.
- Mejoras en las **pruebas** creadas y cubrir más casos.

Imagen: Jorge Arturo Andrade. Pexels

Fin de la presentación.

Muchas gracias por su atención.