

### Tabla de contenidos

1. Objetivos	1
2. Descripción del Caso de Estudio	1
Tarea 1 Pipeline -Unix (40 puntos)	2
Tarea 2 (Bosquejo arquitectónico, 40 puntos)	3
Tarea 3 (Explicar lo realizado y presentar la arquitectura, 20 puntos)	4
Entrega	4
3. Plazos de entrega	5

## 1. Objetivos

El principal objetivo de la práctica es la aplicación de las diferentes actividades estudiadas en la asignatura para dar cobertura a los procesos relacionados con la especificación de la arquitectura de un sistema. Para el desarrollo de la práctica se aplicarán las herramientas de modelado adecuadas.

## 2. Descripción del Caso de Estudio

El objetivo de la parte práctica del curso es aplicar los hallazgos teóricos que se discuten en la conferencia en unas situaciones prácticas. A tal fin los estudiantes desarrollarán una arquitectura de software e implementar un prototipo de arquitectura para un sistema y el desarrollo teórico de una futura arquitectura. El proyecto se lleva a cabo en un equipo de 4 estudiantes.

El lenguaje de programación, la base de datos y el sistema operativo pueden ser elegidos libremente por los participantes. El código debe ejecutarse en un ordenador portátil (o dispositivo móvil), donde cada grupo pueda demostrar el funcionamiento de sus aplicaciones

Por favor, cada equipo debe asegurar de que su código es fácilmente legible, reutilizable y fácil de mantener.

Debido al corto plazo de la corrección repito, el código debe tener los siguientes atributos de calidad:

- ✓ Legibilidad
- ✓ mantenibilidad
- ✓ Facilidad de uso, ...

A título de ejemplo, la aplicación web deber ser utilizable por una persona de edad avanzada , no es un técnico de informática, así que por favor, pensar en la facilidad de uso. Por otra parte, no se debe de tardar más que unos pocos segundos en obtener respuesta a la acción de un usuarios.

El ejercicio se divide en dos tareas separadas, donde la primera tarea debe completarse usando comandos de Unix, y la segunda tarea debe preparar un esquema de arquitectura que soporte funciones interactivas.

# Práctica de Arquitectura Software

## Curso 2018-2019

Versión 1.0

El objetivo final es poder definir la implementación de una aplicación basada en una base de datos abierta de España. Los datos de la aplicación han de tomar de OpenStreetMap (<http://download.geofabrik.de/europe.html>).

### Tarea 1 Pipeline -Unix (40 puntos)

Escribir primero una tubería de comandos Unix (o similar) para importar un dataset a una base de datos, usando el concepto de tubería y herramientas de la línea de comandos. Al final de la ejecución la base de datos debe estar completamente poblada. Los datos deben ser procesados, convertidos a CSV, filtrados y entonces almacenados en la base de datos.

- Descargar OpenStreetMap-datos para España: (spain-latest.osm.bz2 ) Escribir una pipeline de comandos de Unix (no se debe crear un script) que resuelve las tareas siguientes:
  - Utilice osmfilter con el fin de mantener TODOS los nodos-parada (trenes, metro y autobús), por ejemplo `./osmfilter spain-latest.osm --keep = "train = train_stop" -o = train_stop.osm` .
  - Juega un poco con OpenStreetMap un poco para ver lo que tiene que filtrar.
  - Filtrar las paradas que quedan justo dentro de Madrid. Escoger bien el termino municipal o bien el territorio de la Comunidad Autónoma). Para ello, puede utilizar este archivo .poly: ( CiudadMadrid o ComunidadMadrid disponible en moodle)
  - Utilice osmconvert para convertir a formato CSV datos `./osmconvert all_stop.osm --all-to-nodos --csv = "@id @lon @lat name" --csv-headline`. Se debe guardar al menos el identificador, la longitud, la latitud y el nombre de las paradas. Ese es el conjunto mínimo pero sois libres si queréis añadir mas metadatos).
  - Utilizar herramientas estándar del sistema operativo (awk, sed, grep, ...) con el fin de filtrar las paradas que no tienen un nombre (es decir, el nombre está vacío), si los hay.
  - Cargar el contenido a una base de datos donde se pueda consultar con SQL estándar (por ejemplo: mysql)

Utilizando el mismo conjunto de datos que para la primera tubería. Escribe una segunda tubería de comandos de Unix que resuelve las tareas siguientes:

- Utilice osmfilter con el fin de mantener a los siguientes tipos de restaurantes (etiqueta: amenity):
  - Bar
  - Cafe
  - Restaurant
- Juega un poco con OpenStreetMap un poco para ver lo que tiene que filtrar.
- Al igual que en la tubería anterior, mirar a ver si en este caso puedes de guardar las direcciones web de los restaurantes, siempre que la tengan.
- Filtrar por restaurantes dentro de Madrid. Para ello, puede utilizar este archivo .poly: ( Madrid.poly )

# Práctica de Arquitectura Software

## Curso 2018-2019

Versión 1.0

- Utilice osmconvert para convertir a formato CSV. Tienes que guardar al menos el identificador, longitud, latitud y el nombre de las entradas, pero eres libre de utilizar la mayor cantidad de metadatos que desee.
- Utilizar programas de terminal Unix estándar ( como awk, sed, grep, ....) con el fin de filtrar los restaurantes que no tienen un nombre (es decir, el nombre está vacío), si los hay.
- Cargar el contenido a una base de datos que comprende consultas SQL estándar (por ejemplo: mysql)

**La creación de la base de datos y tabla (s) es necesario que ocurra en una de las dos tuberías (lo normal es en la primera).** Por otra parte, las tuberías no deben fallar si la base de datos o las tablas ya existen. Se puede usar un script para crear la infraestructura necesaria base de datos o importar los datos, pero el script tiene que ser ejecutado desde su tubería. Se puede especificar el orden en que las dos tuberías deben ser ejecutadas (es decir, la que se ejecuta en primer lugar crea la base de datos). La decisión sobre el comportamiento en relación con la sobreescritura o de agregar las entradas ya existentes depende de lo que se decida en el grupo. Se puede bien sobrescribir o bien anexar.

Nota: Ese pueden utilizar archivos permanentes si se desea, pero debe haber al menos dos operadores '|' de encadenamiento o bien operadores de redireccionamiento ( '>', '<'). dentro de cada tubería.

Ejemplo de tubería: `cat /proc/cpuinfo | grep -A 5 -B 5MHz | grep 'core id' | awk NR == 1`

## Tarea 2 (Bosquejo arquitectónico, 40 puntos)

Crear un boceto conceptual de una aplicación servirá de apoyo a los medios de transporte en el contexto de las ciudades inteligentes. El boceto representa una arquitectura de una aplicación web donde se pudieran presentar, filtrar selectivamente y editar los datos. Un ejemplo de las funciones a soportar para los datos de las paradas de transporte y los restaurantes de Madrid serian:

- Revisar y filtrar las paradas por nombre, longitud, latitud (por encima o por debajo de una latitud , por la derecha o la izquierda de una longitud). Los tres filtros deberían ser combinables, pero también debería trabajar de manera independiente.
- Comprobar la conexión entre dos paradas. Dados dos nombres de paradas, comprobar si existe una linea que conecta directamente la paradas. Si existe al menos una linea, presentar todas las lineas que conectan las dos paradas directamente.
- Crear nuevas lineas, crear una nueva linea que conecte dos o más paradas existentes.
- La nueva aplicación debería funciona tanto en página web como en terminales móviles.
- Buscador de restaurantes, para una parada dada buscar todos los restaurantes cercanos a una determinada distancia. Para simplificar el problema la distancia puede adecuarse a las dimensiones de un cuadrado en lugar de un círculo.

Aseguraros de que se describen todos los servicios necesarios para la arquitectura. Presentar el bosquejo como un archivo PDF con un máximo de 2 paginas en formato A4. El bosquejo deberá contener al menos 3 funcionalidades diferentes (cada uno se ejecutará en un espacio separado)

# Práctica de Arquitectura Software

## Curso 2018-2019

Versión 1.0

Como alternativa, también puede optar por diseñar un prototipo si es más fácil para vosotros, en lugar del bosquejo arquitectónico.

### **Tarea 3 (Explicar lo realizado y presentar la arquitectura, 20 puntos)**

Para facilitar la corrección será necesario un pequeño fichero de presentación en formato PPT o Key que explique de manera sintética lo realizado en cada tarea.

Preparar una presentación de la arquitectura dirigida al personal de desarrollo y al equipo de marketing de la aplicación.

La presentación deberá prestar especial atención a los siguientes aspectos:

1. Identificación de los objetivos de negocio que dirigirán las decisiones del diseño de la arquitectura.
2. Identificación de las principales partes interesadas (stakeholders) y roles involucrados en la arquitectura.
3. Descripción de la arquitectura justificando las decisiones de diseño que se han tomado durante el proceso de construcción de la arquitectura. La documentación a entregar será la correspondiente a la especificación de la arquitectura mediante artefactos que sigan un formalismo (por ejemplo diagramas de UML) y que contemple algunos los siguientes puntos de vista de acuerdo al público objetivo, según el modelo de Kruchten (4+1): Lógico, Proceso, Implementación, Despliegue y Escenarios.
4. Descripción de la trazabilidad y la consistencia de la arquitectura. En particular se analizará en la corrección de las siguientes relaciones:
  - a. Entre los objetivos de negocio y los escenarios de uso de la arquitectura propuesta.
  - b. Entre los escenarios de uso y los elementos de cada una de las vistas (Lógico, Proceso, Implementación, Despliegue) de la arquitectura propuesta.
  - c. Para cada una de las vistas, comprobar la consistencia de los elementos de una vista con todas las demás

### **Entrega**

La entrega se realizará a través del sistema Moodle.

Debe enviarse un fichero comprimido. La estructura del fichero comprimido se ajustará a una carpeta por tarea de manera clara (por ejemplo, Tarea 1, Tarea 2,... o Tarea01, Tarea02, o Task1, Task2,...).

El alumno podrá elegir el conjunto de herramientas que vaya a utilizar para describir las vistas. Independientemente de la herramienta utilizada, el formato de los documentos que se presente deberá ser Adobe PDF.

### **5. Plazos de entrega**

La práctica se deberá entregar en la fecha prevista en la presentación de la asignatura: Viernes 24 de mayo de 2019. Se recuerda que previamente a la entrega, el alumno puede consultar aquellos aspectos que le planteen dudas al profesor de la asignatura.