

### Tabla de contenidos

1. Objetivos	1
2. Descripción del Caso de Estudio	1
Descripción de la aplicación a diseñar	1
A. Tarea 1 Pipeline -Linea de comandos de Unix (25 puntos)	2
B. Tarea 2 Diseño arquitectónico (40 puntos)	4
C. Tarea 3 Presentación de la arquitectura (35 puntos)	4
3. Entrega	5

## 1. Objetivos

El principal objetivo de la práctica es la aplicación de las diferentes actividades estudiadas en la asignatura para <<aplicarlos en la práctica>>. Para el desarrollo de la práctica se usaran las herramientas de modelado que se estimen más adecuadas.

## 2. Descripción del Caso de Estudio

El objetivo de la parte práctica del curso es aplicar los conocimientos teóricos en unas situaciones prácticas.

A tal fin los grupos de estudiantes desarrollarán una arquitectura de software para un sistema y la implementación de una parte del sistema usando una arquitectura Pipeline y realizaran el diseño de una arquitectura completa.

El proyecto se lleva a cabo en un equipo. El lenguaje de programación, la base de datos y el sistema operativo serán elegidos libremente por los participantes. El código debería poderse ejecutar en un ordenador portátil y el grupo deberá ser capaz de demostrar el funcionamiento si el profesor se lo requiere.

Debido al corto plazo de la corrección repito, el código debe tener los siguientes atributos de calidad: **Legible, reusable y mantenible.**

El ejercicio se divide en tres tareas separadas, donde la primera tarea debe completarse usando comandos de Unix, la segunda tarea debe preparar un esquema de arquitectura que soporte funciones interactivas. Y la tercera es la correspondiente documentación de los realizado.

---

### Descripción de la aplicación a diseñar

La meta del ejercicio sea implementar una aplicación basada en base de datos, aplicando arquitecturas.

Los datos de la aplicación se toman de OpenStreetMap, y se deberán filtrar para las cafes, bares, pubs, tabernas, restaurantes y establecimientos de comida rápida de Madrid. El usuario debería de ser capaz de encontrar a cualquiera de ellos, bien por unas coordenadas que establece una cuadrado delimitador y/o por una categoría específica (e.j. "bar"). El punto central del cuadrado se encuentra especificada por las coordenadas dadas por el usuario y el tamaño del cuadrado debe ser ajustable por el usuario (por ejemplo, 100x100m).

# Práctica de Arquitectura Software

## Curso 2020-2021

Versión 1.0

Además se debe permitir al usuario la posibilidad de añadir, editar o eliminar de manera individual, restaurantes, cafes, bars, pubs y establecimientos de comida rápida (“fast food”).

Las tareas a realizar son:

1. Arquitectura Pipeline (estilo de comandos de Unix)
2. Diseño arquitectónico
3. Presentación de la arquitectura diseñada a un reducido número de partes interesadas

Las bases de datos a manejar son:

- Mapa OpenStreetMap de España (descargado de <http://download.geofabrik.de/europe.html>)
- Fichero de Puntos de Interés de Madrid: POI\_Madrid.csv
- Limites de Madrid: ComunidadMadrid.poly (si fuera necesario se puede trabajar solo dentro de la ciudad de Madrid usando el fichero CiudadMadrid.poly)

### A. Tarea 1 Pipeline -Linea de comandos de Unix (25 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 extraer TODOS los nodos (cafes, bars, pubs, restaurants y fast foods)
  - Filtrar aquellos que quedan justo dentro de Madrid. Escoger bien el termino municipal de Madrid (se reduce más el tamaño de los datos) 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_amenities.osm --all-to-nodos --csv = "@id @lon @lat name opening_hours" --csv-headline`. Se debe guardar al menos el identificador, la longitud, la latitud, nombre y el horario de apertura. (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 entradas 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: sqlite3)

La creación de la base de datos y de las tablas también necesita que se genere en la “pipeline”. Además, el pipeline no debería fallar si la base de datos o las tabla(s) ya existieran. Tu puedes

# Práctica de Arquitectura Software

## Curso 2020-2021

Versión 1.0

usar un cripta para crear la infraestructura de base de datos o importar los datos, pero el script necesita ser ejecutado desde el pipeline.

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

- Utilice `osmfilter` con el fin de mantener a los siguientes tipos de restaurants (etiqueta: amenity=restaurant) de acuerdo al tipo de cocina:
  - `cuisine=regional`
  - `cuisine=mexican`
  - `cuisine=japanese`
- 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 el teléfono y la dirección web de los restaurants, siempre que la tengan.
  - Filtrar por restaurants dentro de Madrid. Para ello, puede utilizar este archivo `.poly`: ( `Madrid.poly` )
  - Utilizar `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 cafes 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

**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 sobrescritura 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`

### B. Tarea 2 Diseño arquitectónico (40 puntos)

Crear un diseño arquitectónico de una aplicación servirá de apoyo al turismo en el contexto de las ciudades inteligentes.

El diseño 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 lugares turísticos, y pequeños comercios como son los cafes, bares, tabernas, pubs y establecimientos de restauración de Madrid serian:

- Listar y filtrar los establecimientos por categoría, nombre, alrededor de un punto de interés (POI) seleccionado por el usuario (por ejemplo, 100 m), (10 puntos)
- Salvar una búsqueda como un favorito con un nombre descriptivo. Por ejemplo “Cerca de la plaza del Callao”) (10 puntos)
- Editar/borrar un favorito (10 puntos)
- Añadir/editar/borrar un cafe, bar, taberna, pub o establecimiento de comida(10 puntos)

Aseguraros de que se describen todos los servicios necesarios para la arquitectura. Presentar el diseño como un archivo PDF con un máximo de 3 paginas en formato A4. El diseño deberá contener al menos 3 funcionalidades diferentes.

### C. Tarea 3 Presentación de la arquitectura (35 puntos)

Esta tarea tendrán dos partes claramente separadas:

- A. Un fichero de presentación donde explique el equipo de manera sintética el esfuerzo dedicado y los resultados en cada tarea, haciendo hincapié en lo conseguido y también en las dificultades surgidas
- B. Preparar un segundo fichero de presentación de la arquitectura para una reunión conjunta con el **jefe de proyecto y el equipo de marketing** de la aplicación.

La presentación de la arquitectura deberá prestar especial atención a los siguientes aspectos:

1. Identificación de las características principales e intereses de las partes interesadas (stakeholders) y roles involucrados en la arquitectura. Indicar los materiales seleccionados de acuerdo a la audiencia prevista.
2. Documentación de la arquitectura justificando las decisiones de diseño que se han tomado durante el proceso de diseño de la arquitectura, desde la Identificación de los drivers de la arquitectura.
  - a. La documentación a entregar será la correspondiente a la especificación de la arquitectura realizada en el paso anterior asegurando que que sigan un formalismo (por ejemplo diagramas de UML) y que contemple algunos de los siguientes puntos de vista de acuerdo al publico objetivo, de acuerdo al modelo de Kruchten (4+1): Lógico, Proceso, Implementación, Despliegue y Escenarios.
  - b. La documentación de textual de la arquitectura, se aportará un fichero word separado. Teniendo en cuenta que siempre se ha de mantener la consistencia entre el fichero word y el material de la presentación.
3. Demostración de la consistencia de la arquitectura. En particular se dedicará espacio para presentar la trazabilidad mediante tablas, donde se pueda seguir la pista de los componentes a lo largo de las vistas presentadas.

# Práctica de Arquitectura Software

## Curso 2020-2021

Versión 1.0

Para facilitar la corrección será necesario tanto el fichero de presentación de la parte A como de la parte B, se generen o bien en formato bien .PPT o bien .Key

### 3. Entrega

La entrega se realizará a través del sistema Moodle.

Debe enviarse un fichero comprimido (.zip o similar) a la plataforma Moodle. 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 equipo 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 PDF.

La práctica se deberá entregar en la fecha prevista en la plataforma Moodle. Se recuerda que previamente a la entrega, los equipos puede consultar por email o via Teams aquellos aspectos que le planteen dudas al profesor de la asignatura.