Universidad Politécnica de Cartagena

Escuela Técnica Superior de Ingeniería de Telecomunicación

PRÁCTICAS DE APLICACIONES EN INTERNET

Propuesta del Trabajo de Prácticas 2025/2026

Aplicación web de puntuación y recomendación de restaurantes

Revisión 0.1

Profesores:
Esteban Egea López

Índice.

1 Consideraciones generales.

1.1 Objetivos

En este trabajo de prácticas los alumnos realizarán por parejas una aplicación de puntuación y recomendación de restaurantes. El objetivo es que el alumno se familiarice con las tecnologías web más empleadas en la actualidad (HTML, CSS, PHP, javascript) así como con el tratamiento de los datos y los algoritmos de análisis de información más relevantes.

1.2 Introducción

La aplicación a desarrollar imitará de manera simplificada el funcionamiento de aplicaciones como http://www.yelp.es, por ejemplo. Se trata de una aplicación que dispone de un catálogo de restaurantes, con información variada sobre los mismos, como fotos, localización, horario, que cualquier usuario puede consultar. La aplicación permite a los usuarios registrarse. Los usuarios registrados pueden puntuar los restaurantes, añadir comentarios y recibir una recomendación del sistema.
Para poder realizar la aplicación se usará el *dataset* de Yelp Open Dataset (https://business.yelp.com/data/resources/open-dataset/). El dataset contiene más de 150 mil negocios, pero se ha extraído solo la información correspondiente a los restaurantes de Philadelphia (EEUU). Además, se dispone de las fotos asociadas a los restaurantes. Esta información base se pondrá a disposición de los alumnos para que sobre ella implementen su aplicación, tal y como se describe en los apartados posteriores. Toda la información se proporciona en una BBDD disponible en su cuenta del laboratorio, que **pueden modificar si es necesario para su aplicación**.

1.2.1 Qué deben hacer los alumnos

Los alumnos deben implementar por parejas una aplicación web con la funcionalidad que se especifica en este documento. Para ello deberán:

1. Programar los scripts en PHP con la lógica de la aplicación.
2. Programar cualquier fichero adicional con código HTML, CSS o javascript necesario para la aplicación.
3. Programar los scripts de Python con el algoritmo de análisis de datos.

La aplicación debe ser completamente funcional y debe poder probarse. Para ello, se alojará la aplicación en la cuenta del servidor del laboratorio (labit601.upct.es) y se accederá a ella a través del servidor HTTP del laboratorio.

Se deberá entregar una memoria del trabajo realizado. El contenido de la memoria y los criterios de evaluación también están descritos en esta propuesta.

1.2.2 Cómo está organizada esta propuesta.

En el apartado 2 se describe la funcionalidad mínima a implementar y las partes en las que se ha dividido el desarrollo, especificando cuáles serán opcionales.
En el apartado 3 se especifican los criterios de evaluación y la memoria y el material que se debe entregar.
Finalmente, en el apartado 4 se dan algunas sugerencias sobre la implementación y el uso de librerías adicionales.

2 Desarrollo de la aplicación

La **funcionalidad mínima** que debe proporcionar la aplicación es la siguiente:

1) **Catálogo**. Mostrar y permitir la navegación por todo el catálogo de restaurantes. Para ello debe mostrar:
   a) Al menos una imagen del restaurante (si está disponible)
   b) Nombre del restaurante hiperenlazado a una nueva página que mostrará el contenido descrito en 4 (ver abajo).
   c) Puntuación media del restaurante (stars), número de puntuaciones recibidas y **puntuación ponderada del restaurante.**
   d) Se debe permitir la **reordenación** de los elementos en base, al menos, a los campos nombre y puntuación ponderada.
2) **Login de usuario**. Mostrar un enlace o un formulario en la página de inicio que permita iniciar una sesión a los usuarios registrados o registrar un nuevo usuario.
3) **Registro de usuario**. Permite añadir un nuevo usuario. Los datos que puede aportar el usuario son el nombre, tal y como está la BBDD que se le proporciona, pero puede incluir nuevos campos si modifica la BBDD original.

4) **Registro de nuevo negocio**. Este formulario permitirá añadir un nuevo negocio al sistema. No es necesario que lo haga un usuario registrado, ni vincular el negocio a ningún usuario. Para registrar un negocio hay que aportar al menos:
   a) Nombre
   b) Ciudad
   c) Estado (Región, Comunidad Autónoma)
   d) Latitud y Longitud
   e) Categorías
   f) **Al menos una foto del negocio**
5) **Restaurante**. Mostrar los campos asociados al negocio (ver 1) que considere oportuno y:
   a) Las categorías a las que pertenece.
   b) Los comentarios recibidos con el nombre del usuario que lo realizó.
   c) Puntuación recibida por el usuario (si la ha puntuado ya) si el usuario se ha identificado y ha iniciado una sesión.
   d) Posibilidad de puntuar o cambiar la puntación si el usuario se ha identificado y ha iniciado una sesión.
   e) Posibilidad de añadir un comentario si el usuario se ha identificado y ha iniciado una sesión.
6) **Usuario**. Mostrar los datos personales del usuario y permitir
   a) **Generar recomendaciones personalizadas**. Ejecutará el algoritmo de recomendación y generará una puntuación recomendada para cada uno de los restaurantes del catálogo.
   b) **Mostrar las recomendaciones generadas**. Se mostrarán o bien todas o bien un subconjunto, ordenadas por mayor puntuación.
   c) Modificar datos personales (en caso de que se hayan incorporado datos adicionales a los presentes en la tabla al usuario).
7) **Formato**. Debe proporcionar al menos una hoja de estilo que dé formato a su aplicación. El formato es libre pero se valorará un uso adecuado.
8) **Contenido dinámico**. Debe utilizar, donde considere, javascript para generar el contenido, formato o responder a acciones de usuario. Por ejemplo, puede usarlo para validar los campos del formulario de registro o para asignar las puntuaciones a un restaurante. En la memoria, indique cómo ha usado javascript.

Para implementar esta funcionalidad se le proporcionará una base de datos ya rellena con los **datos mínimos** que necesita. La base de datos se describe en 2.2. Para generar las recomendaciones deberá utilizar el algoritmo de filtrado colaborativo que realizó en prácticas. Se le proporcionará una interfaz con el servidor Python que lo ejecutará, como se describe en 2.3. La parte fundamental de la aplicación es la lógica de aplicación en PHP, que se describe en 2.1.
La **calificación** del trabajo se reparte entre una **parte obligatoria con la funcionalidad mínima (75%), y una de funcionalidad opcional (25%)**.

2.1 Lógica de la aplicación

La lógica de la aplicación.
Se implementará mediante **scripts PHP** que proporcionarán *al menos* la **funcionalidad mínima** requerida descrita en el apartado anterior. Organice la funcionalidad en scripts separados y utilice funciones para reutilizar partes de código que se repitan. Puede utilizar *require* y *require_once* para cargar librerías de funciones u otros scripts PHP. En particular, considere la posibilidad de agrupar el HTML común a todas las páginas en funciones/scripts de cabecera o pie.

Imágenes
Las imágenes utilizadas se han comprimido en un fichero disponible en http://labit601.upct.es/~eegea/yelp/pics/. En esta localización puede encontrar el fichero phil-photos.zip (773 MB) o puede directamente enlazar a las fotos que se encuentran en esa localización (tenga en cuenta que al servidor de labit601.upct.es solo se puede acceder desde fuera de la red de la UPCT con la VPN).

Interfaz de consultas SQL.
La lógica de la aplicación realiza consultas a la base de datos mediante SQL, insertando las variables PHP necesarias. Para realizar la funcionalidad mínima, debe desarrollar un conjunto de consultas SQL que le permitan recuperar la información necesaria de la BBDD. Es una buena práctica separar las consultas del resto de la lógica y agruparlas en un único fichero PHP.

Estilo y javascript.
Desarrolle una hoja de estilo y código javascript para implementar la funcionalidad o el estilo.

Puntuación ponderada: Bayesian Ranking

Este tipo de aplicaciones presentan el problema de cómo agregar puntuaciones para poder comparar una película con otra. Por ejemplo, ¿está mejor valorada un restaurante con 356 votos y media 3.8 que uno con 10 votos y media 4.8? La respuesta a esta pregunta es relativamente compleja y puede encontrar una discusión muy interesante al respecto en [1, capítulo 5]. Si la lee recibirá una explicación sobre el método de ponderación que va a utilizar en este trabajo y que se describe a continuación. Para obtener una **puntuación ponderada**, se aplicará la siguiente fórmula a cada restaurante *i*:

$$p_i = \frac{NR + n_i r_i}{N + n_i}$$

donde N es el número total de restaurantes, R es la puntuación media de todos los restaurantes, n_i es el número de puntuaciones del restaurante *i* y r_i es la puntuación media del restaurante *i*.

MATERIAL A ENTREGAR:

- **Scripts PHP, hojas de estilo, ficheros con javascript, HTML y scripts python.** Estarán localizados en su directorio *public_html* bajo el directorio *video*, de manera que la aplicación pueda ser probada.
- **Memoria** descriptiva de la lógica de la aplicación con una justificación de las decisiones de diseño. **No incluya el código completo**. Limítese a describir cómo se ha organizado la lógica, dónde se ha implementado la funcionalidad y comente las decisiones de diseño que crea más relevantes.

2.2 Diseño de la base de datos

Dispone de una base de datos con los datos necesarios para la aplicación. Para trabajar con ella, cada pareja cuenta con un usuario de la base de datos y una base de datos para desarrollo. Se le asignará un nombre de usuario y clave para trabajar con ellos. Puede utilizar phpmyadmin (http://labit601.upct.es/phpmyadmin/) o directamente el comando *mysql* para trabajar con su base de datos. Si realiza el trabajo localmente, deberá exportar la BBDD del servidor del laboratorio, e importarla en su equipo. Para ello, use phpmyadmin y exporte a un fichero SQL su base de datos. Luego use ese fichero para importarla en su equipo local. También está disponible la base de datos en Aula Virtual.

La estructura de las tablas de la base de datos la puede comprobar mediante phpmyadmin. No se ha establecido ninguna relación entre tablas, lo que quiere decir que, aunque hay campos que obviamente hacen referencia a campos de otras tablas, la base de datos **no comprobará la integridad referencial automáticamente**.

Puede modificar y añadir las tablas y campos que considere necesarios para implementar su aplicación. La BBDD que se le ha proporcionado contiene los datos y tablas **mínimos** para desarrollar el trabajo.

Finalmente, observe que **la tabla para guardar las recomendaciones no está incluida, deberá crearla según las necesidades del algoritmo** (ver Sección 2.3).

2.3 Interfaz con los algoritmos de análisis de la aplicación

Los algoritmos de análisis (recomendación, etc.) se implementarán mediante Python, por lo que es necesaria una interfaz entre PHP y Python. En una aplicación real de este estilo es muy poco probable que se hiciera de esta forma, sino que se implementarían los algoritmos en un lenguaje más apropiado para su uso en un entorno de producción. En nuestro caso, para simplificar el desarrollo utilizará directamente Python.

Para ello haremos lo siguiente:

1. Python se invocará mediante un servidor que se ejecuta en el laboratorio. El servidor recibirá peticiones TCP al puerto 4450, de un script PHP que le enviará el **path absoluto (directorio) en el que se encuentran los scripts de Python necesarios,** así como **la función a ejecutar** incluyendo los parámetros necesarios. **No tiene que implementar este servidor TCP, ya se encuentra funcionando en el laboratorio.** El servidor invocará el script de python correspondiente al algoritmo. **En el aula virtual dispone de *server.py* con el código** de un servidor Python igual al del laboratorio, que puede usar **para probar esta parte en su propio ordenador,** sin necesidad de usar el del laboratorio.
2. El algoritmo obtendrá los datos necesarios directamente de la BBDD mediante consultas SQL y actualizará la BBDD con los datos generados.

Dispondrá en Aula Virtual de los siguientes ficheros:

- **Un script PHP** que se conecte mediante un socket TCP al servidor de Python y le pase:
    - la **ruta absoluta de los scripts de Python** que tiene que ejecutar

    - y el **nombre de la función que tiene que ejecutar**. Esa función es precisamente la que implementa el algoritmo de filtrado colaborativo. Observe que el nombre de la función debe pasarse con el nombre del script de Python que la contiene. Es decir, si la función se llama *recomendar* y está dentro del fichero *rec.py* hay que pasarla como *rec.recomendar(25)*, por ejemplo.
- Dos scripts de Python que implementan: **una función *getData()*** que genera las matrices R, Y y movieList a partir de las tablas de la base de datos. Además, en la práctica los resultados del algoritmo simplemente se muestran por pantalla. Pero en la aplicación, los datos se tienen que guardar en la BBDD, mediante la **función *updateRecommendation()*.** Estas funciones, por tanto, se ejecutan como parte del código de la función que implementa el algoritmo de filtrado colaborativo.

La función Python que implementa el algoritmo contendrá el **código modificado** de la práctica de Sistemas de Recomendación. **Tiene que implementar la función de Python *recomendar*(*userid*)**: esta función, como decimos contendrá el código modificado de la práctica de Sistemas de Recomendación que implementa el algoritmo de filtrado colaborativo (ver apartado siguiente).
El servidor espera recibir:

1. Una cadena de caracteres con la ruta absoluta al directorio en el que se encuentran sus scripts de Python, terminada con retorno de carro. Por ejemplo, en PHP tendría que establecer un socket con el servidor y enviar una variable así *$ruta="/home/alumnos/ai/python\r\n";*
2. Una cadena de caracteres con la invocación de la función que ejecutará su algoritmo y el fichero en el que se incluye. Por ejemplo, si su función se ha declarado en el script *rec.py* como *recomendar(id*), es decir, se le pasa como parámetro un identificador de usuario, deberá usar una variable en PHP *$fun="rec.recomendar(".$userid.")\r\n"*.

Observe como se introduce el "\r\n" al final de las cadenas para que el servidor separe los dos campos, r. Finalmente, antes de escribir dichas cadenas sobre su socket, concatena con *chr(0)* para asegurar que se fuerza el envío y se recibe correctamente. Con el ejemplo anterior, tendríamos que hacer

```
$info = $ruta.$fun.$chr(0);
$sent=socket_write($socket, $info, strlen($info));
```

El servidor TCP de Python ejecutándose en el servido del laboratorio escucha el **puerto 4450**.

2.4 Algoritmo de filtrado colaborativo y recomendación

Debe incorporar en la aplicación el código realizado en la práctica de Sistemas de Recomendación basada en filtrado colaborativo. Para obtener las matrices R e Y puede **emplear la función *getData()* del apartado anterior**. Después de entrenar el algoritmo y generar las puntuaciones (mediante el código realizado en la práctica), debe actualizarlas en la base de datos mediante la **función *updateRecommendation()* del apartado anterior**.
Entonces, tiene que realizar lo siguiente:

- Una función de Python llamada ***recomendar*(*userid*)**. Esta función es la que implementará el algoritmo de filtrado colaborativo y actualizará una nueva tabla de la base de datos con las películas recomendadas. **El número de características que usará el algoritmo de filtrado colaborativo puede variar**.
- **Crear una nueva tabla** en la base de datos para almacenar las recomendaciones.

2.5 Funcionalidad opcional

Un porcentaje de la nota de este trabajo (**25%**) se asignará a la innovación o extensión de la aplicación aquí descrita.

Se proponen varias extensiones (solo es necesario implementar una de ellas para conseguir la nota):

- **Utilizar librerías de geolocalización** en Javascript para mostrar los restaurantes sobre un mapa interactivo. Puede usar las librerías de leaflet (https://leafletjs.com/) o Google Maps (https://developers.google.com/maps). O cualquier otra. También existen librerías llamadas "reverse geocoding" que le permiten obtener una dirección a partir de su lat/lon, ya que esta información no está disponible en el dataset. Sin embargo, tenga en cuenta que en muchos casos es necesaria una suscripción.
- **"A otros usuarios le gusta":** Extender el algoritmo de filtrado colaborativo para la aplicación muestre los restaurantes que mejor ha puntuado el usuario más parecido al usuario actual.
- **"Restaurantes similares":** cuando el usuario ve la descripción de un restaurante, añadir un botón adicional para mostrar restaurantes similares. Para ello use el algoritmo de filtrado colaborativo para buscar restaurantes similares al que se está examinando.

- **Utilizar librerías de algoritmos de recomendación** de Python alternativas. Hay multitud de librerías en Python que ejecutan algoritmos de recomendación, como Surprise o RecBole. Una vez se tienen los datos en el formato adecuado, usarlas es inmediato.

3 Material a entregar y criterios de evaluación

Los alumnos deberán entregar una **memoria breve** del trabajo, además de **alojar el código** de la aplicación en su cuenta del laboratorio, de manera que quede accesible mediante el servidor HTTP. Se avisará con suficiente antelación de la fecha límite de entrega. Las **memorias** se subirán como entregable al **aula virtual**, por un único miembro del grupo. **No se admitirá ningún trabajo en fechas posteriores a la fecha límite**. Es posible que en algún caso los profesores necesiten reunirse con los alumnos de un grupo para evaluar su trabajo. Para esos casos se publicará una lista de los grupos que deben entrevistarse con los profesores para explicar su trabajo.

3.1 Memoria y Código

La **memoria** debe contener:

- Nombre completo de los componentes del grupo, correo electrónico y, al menos, un teléfono de contacto.
- Índice
- URL de acceso a la aplicación.
- La memoria descrita en el apartado 2.1, describiendo la implementación de la aplicación.
- Una lista con todos los scripts PHP implementados, describiendo la funcionalidad que implementan.
- Una lista con todos los ficheros adicionales que haya utilizado (CSS, javascript), describiendo la funcionalidad que implementan.
- Si ha utilizado un framework para desarrollo PHP, CSS o Javascript, una descripción del uso que hace.
- Una descripción de cualquier funcionalidad adicional implementada.

El código se alojará en el directorio *public_html*, bajo un directorio *video* para que sea accesible al servidor HTTP del laboratorio, y **deberá ser completamente funcional, es decir, debe poder probarse la aplicación mediante un navegador**.

3.2 Criterios de evaluación

Aunque se ha mencionado antes, se resumen de nuevo los criterios de evaluación. La puntuación total del trabajo se ha repartido entre las distintas partes:

Apartado	Puntuación máxima
Funcionalidad mínima	7.5
Opcional	2.5

Para la evaluación se tendrá en cuenta:

- Que se hayan implementado correctamente las funcionalidades necesarias.
- Se valorará la interpretación dada de los resultados del algoritmo de recomendación y se apreciará especialmente que los alumnos demuestren haber consultado bibliografía, Internet, etc, para mejorar sus conclusiones.
- Se valorará la claridad, rigor y concisión en las explicaciones aportadas en la memoria.
- Se valorará la claridad y orden en el código.
- Se valorará que se haya separado la funcionalidad en ficheros y funciones apropiadamente.
- El uso de estilos CSS. Aunque la calidad estética en sí de la aplicación es subjetiva y no se evaluará como tal, sí que se evaluará el uso adecuado de hojas de estilos y técnicas asociadas.

Se valorarán las extensiones de la funcionalidad y/o propuestas de otros algoritmos y el estudio de la influencia de distintos parámetros en el rendimiento. Las aportaciones se valorarán por su **originalidad, su calidad**, y su argumentación, **no por su cantidad ni su extensión**. Estas contribuciones pueden ser útiles para mejorar la **nota final de prácticas** de la asignatura. El **uso de IA generativa no está permitido** más allá de alguna consulta puntual. Si se

determina que se ha usado extensivamente para realizar el trabajo, se convocará al grupo a una reunión con el profesor para determinar si se han cumplido los resultados de aprendizaje.

3.3 Grupos de una persona

Al inicio de esta propuesta se indica que el trabajo debe realizarse en parejas, pero en ocasiones las circunstancias impiden formar o mantener un grupo y se queda una sola persona. Como no es justo aplicar los mismos criterios de evaluación, en estos casos se aplicará el siguiente reparto de puntos:

Grupos de una persona:

Apartado	Puntuación máxima
Funcionalidad mínima	8.5
Opcional	1.5

4 Notas adicionales para la implementación

4.1 Implementación de la lógica en PHP

- Se recomienda realizar la implementación de la lógica de manera incremental por unidades funcionales. Por ejemplo, puede comenzar realizando el/los scripts que permiten la visualización y navegado por el catálogo de restaurantes. A continuación, el/los scripts que permiten ver los detalles de un restaurante y comentarlo. Y así sucesivamente.
- Puede dejar el formato y estilo de su aplicación para el final, pero se recomienda que antes de comenzar la implementación decida al menos un *layout* u organización visual aproximada. De esa manera podrá agrupar en bloques los elementos y realizar unas cabeceras y pies comunes que se mostrarán en todas las páginas. Por ejemplo, puede decidir que su aplicación solo contendrá un bloque de cabecera y otro de contenidos.

4.2 Diseño alternativo de la base de datos

- Si considera que el diseño de la base de datos que se le proporciona no es adecuado, es libre de utilizar un diseño alternativo o añadir las tablas que considere necesario. Deberá describir y justificar brevemente su diseño alternativo en la memoria.

4.3 Uso de frameworks y librerías adicionales

- Si el alumno lo prefiere, se permite el desarrollo de la lógica de la aplicación mediante *frameworks* de desarrollo, como pueden ser *CodeIgniter*, *Zend*, o *CakePHP*, siempre y cuando estén basados en PHP. **SU USO ESTÁ SUJETO A APROBACION POR PARTE DE LOS PROFESORES**. Deberá enviar un correo electrónico al profesor Esteban Egea indicando el *framework* o librerías que se pretende utilizar. Tenga en cuenta que en algunos casos su uso puede ser contraproducente. Tenga en cuenta también que algunos *frameworks* proporcionan mecanismos específicos para la persistencia de datos, por lo que deberá describir en la memoria también la estructura de la base de datos en ese caso.
- Si se utiliza un *framework* la memoria debe incluir una además una descripción de: **instalación, configuración** y **funcionamiento y forma de uso** del *framework* así como una descripción de cómo se ha implementado la lógica de la aplicación en el contexto del framework.
- Se permite también el uso de **librerías para estilo y formato**, como puedan ser *bootstrap* u otras, pero deberá **documentarse su uso** en la memoria.

5 Bibliografía

[1] Mung Chiang. ”Networked Life. 20 Questions and Answers”, Cambridge University Press, Cambridge, UK, 2012