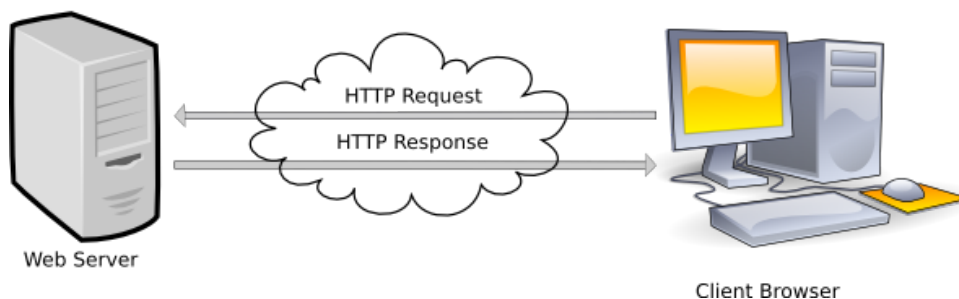


Instituto Tecnológico de Costa Rica
IC-1801 Taller de Programación
Semestre I, 2016

III Tarea Programada

HTTP Request/Response

HTTP (en español *protocolo de transferencia de hipertexto*) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force. Está basado en el modelo solicitud / respuesta, de modo que hay dos tipos de mensajes HTTP: la solicitud y la respuesta. Es uno de los métodos más básicos que las computadoras usan para comunicarse entre sí, donde una máquina envía una solicitud (request) de datos y otra máquina responde (response) la solicitud. Navegar por una página web es un ejemplo de comunicación request/response.



Para mayor información con respecto a este tema puede acceder a los siguientes recursos:

- <https://docs.python.org/3/library/http.client.html>
- http://www.tutorialspoint.com/http/http_tutorial.pdf

XML

Es un estándar ampliamente soportado para describir datos. XML es comúnmente usado para intercambiar datos entre aplicaciones sobre internet. Permite crear marcas para virtualmente cualquier tipo de información, lo cual posibilita la creación de nuevos lenguajes de marcas para describir cualquier tipo de datos, como fórmulas matemáticas, música, noticias, recetas, reportes financieros, entre muchos otros.



Una de las características más importantes de XML es que describe los datos de forma tal que sean entendibles tanto para los humanos como para las computadoras.

A continuación se detalla un ejemplo de un documento XML:

```
<Books>
  <Book ISBN="0553212419">
    <title>Sherlock Holmes: Complete Novels...
    <author>Sir Arthur Conan Doyle</author>
  </Book>
  <Book ISBN="0743273567">
    <title>The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
  </Book>
  <Book ISBN="0684826976">
    <title>Undaunted Courage</title>
    <author>Stephen E. Ambrose</author>
  </Book>
  <Book ISBN="0743203178">
    <title>Nothing Like It In the World</title>
    <author>Stephen E. Ambrose</author>
  </Book>
</Books>
```

Todo documento XML está compuesto de elementos que especifican la estructura del documento. Algunas de sus características son las siguientes:

- Los documentos XML delimitan los elementos con marcas o etiquetas de inicio y fin. Una marca de inicio consiste del nombre del elemento entre corchetes angulares. Ejemplo: **<author>**
- Una marca de cierre consiste del nombre del elemento precedido por un forward slash (/) entre los corchetes angulares.
Ejemplo: **</author>**
- Las etiquetas inicial y final de un elemento encierran el texto que representa los datos
Ejemplo: **<author>Stephen E. Ambrose</author>**
- Cada documento XML debe tener exactamente un elemento raíz que contiene todos los demás elementos
Ejemplo: **<Books>**

Para mayor información con respecto a este tema puede acceder al siguiente recurso: <http://www.w3schools.com/xml/>

SMTP (Envío de correos)

El correo electrónico es considerado el servicio más utilizado de Internet. Por lo tanto, la serie de protocolos TCP/IP ofrece una gama de protocolos que permiten una fácil administración del enrutamiento del correo electrónico a través de la red.

El **protocolo SMTP** (*Protocolo simple de transferencia de correo*) es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto. Éste es un protocolo que funciona en línea, encapsulado en una trama TCP/IP. El correo se envía directamente al servidor de correo del destinatario.

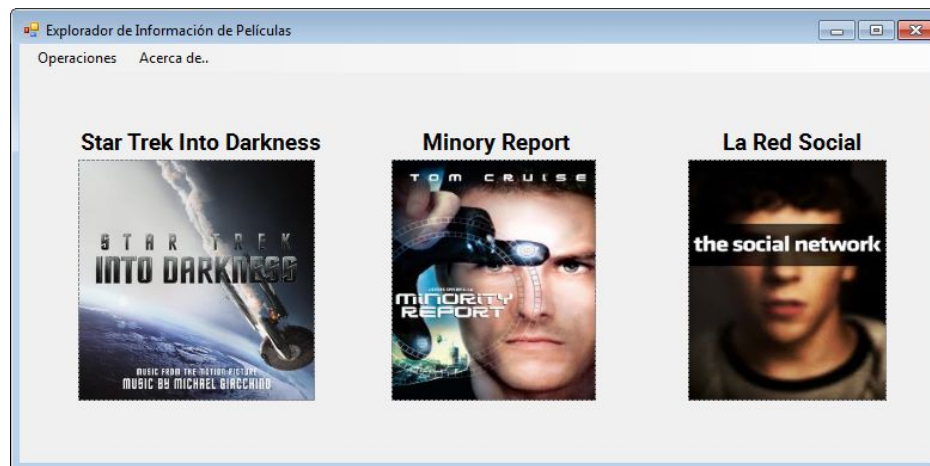


Para mayor información con respecto a este tema puede acceder a los siguientes recursos:

- <https://www.chilkatsoft.com/python-email.asp>
- <https://support.google.com/a/answer/176600?hl=es>

Por hacer:

Implementar una solución computacional con interfaz gráfica de usuario que consiste en un explorador de información de películas, la misma debe mostrar inicialmente una ventana similar a la siguiente:



Las películas que se muestran son solamente de muestra.

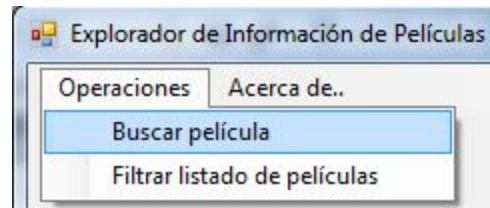
En la pantalla inicial deben desplegarse las películas que el usuario de la aplicación ha consultado satisfactoriamente (mediante la lectura del archivo XML), caso contrario, la pantalla aparecerá en blanco.

Debe considerar la incorporación de la barra de scrolling vertical para que el usuario pueda acceder a cualquier de las películas en la lista. Así mismo se recomienda buscar alguna librería que facilita el manejo de las imágenes, a fin de crear una versión en miniatura (thumbnail) que no debe exceder de 150 x 150 pixeles.

Operaciones del menú

● Operación: Búsqueda de película.

Esta operación del menú permite que el usuario busque la película de su interés en la base de datos de [OMDbi](#). La búsqueda se debe realizar por medio del **nombre de la película**. Utilice XML como protocolo de datos para procesar las respuestas del servidor OMDbi.



Para soportar esta funcionalidad debe mostrar en pantalla una pequeña ventana donde el usuario ingresa el nombre de la película que desea buscar y finalmente **procesar** la respuesta del servidor a fin de determinar si la película existe o no. En caso que el XML indique que no existe la película, debe indicarlo al usuario. Pero en caso contrario debe agregar a una lista el objeto película.

Es sumamente importante mencionar que a fin de dar persistencia en almacenamiento primario, debe guardar la información de la película en un archivo XML en el directorio local de su aplicación.

La etiqueta inicial o raíz de su documento XML debe ser:

```
<películas>  
....  
</películas>
```

Dentro de esta etiqueta es que debe colocar cada una de las películas de acuerdo a los atributos que retorna el servicio OMDbi. Se recomienda instalar alguna librería que permite parsear el contenido de la respuesta del servidor. Por ejemplo: [Chilkat](#)

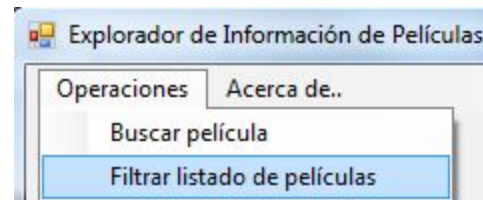
Puede encontrar ejemplos de la librería Chilkat en Python en el siguiente vínculo: <https://www.example-code.com/python/xml.asp>

Debe tomar en cuenta que los datos que su solicitud http debe contener son:

- el nombre de la película que desea consultar,
- el modo de consulta (mismo que debe ser completo) y
- el formato de retorno de la respuesta (formato XML).

● **Operación:** Filtrar listado de películas.

Otra de las opciones del menú de operaciones consiste en filtrar la lista de películas que se despliega en pantalla.

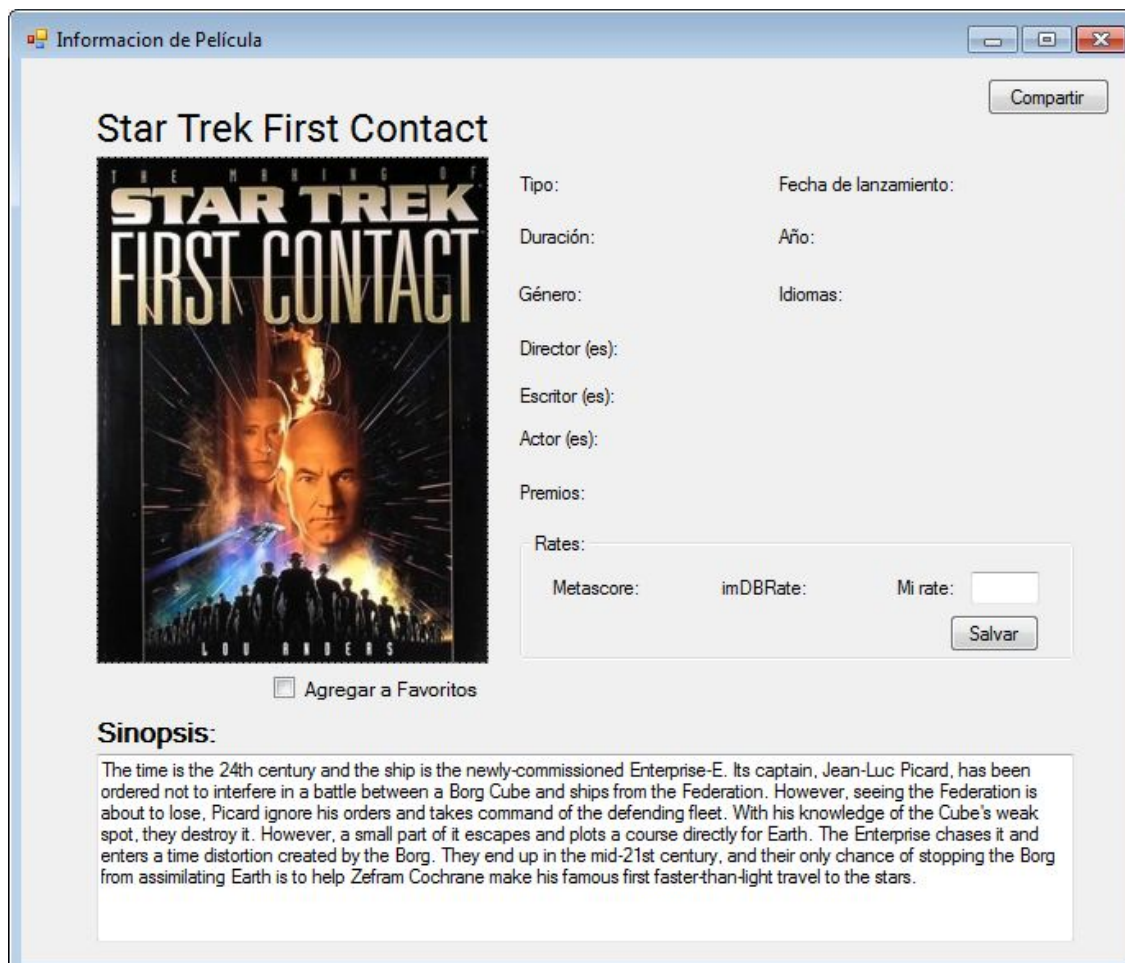


Para esta funcionalidad debe activar o mostrar una pequeña ventana que permite al usuario filtrar la lista de películas de acuerdo a los siguientes criterios:

- Nombre de la película
- Género
- Actor

Operación: Acceder a Información de una película

Cuando el usuario haga click en la imagen de la película debe abrirse una ventana que contenga la siguiente información:



Toda la información anterior, debe obtenerla a partir del objeto película que reside en memoria dentro de una estructura de datos tipo lista. Excepto su propio rate, en caso de no haberlo asignado.

Las funcionalidades de esta sección son las siguientes:

- **Botón de Compartir**

Esta funcionalidad debe soportar el envío de un correo electrónico a la dirección de correo de un amigo. Debe agregar toda la información que se conoce de la película así como adjuntar una imagen.

Considere el uso de la librería [Chilkat](#) para el envío de correo electrónico.

- **Mi rate (Puntuación del usuario)**

Consiste en asignar una puntuación personal del usuario que puede tomar valores desde 1.0 hasta 10.0

En caso que el usuario decida rankear la película este dato se debe reflejar en el objeto película correspondiente y también en el archivo XML que guarda la información de todas las películas que se han consultado.

Consideraciones importantes

- En caso que la película no tenga una dirección URL asociada a su póster, debe utilizar una imagen genérica creada por usted que dice “Imagen no disponible”. Esto aplica para la imagen en miniatura y la imagen ampliada de la película. Se recomienda descargar la imagen original y crear una versión en miniatura para su despliegue en el listado de películas.
- La selección de los nombres de las etiquetas queda a discreción de cada grupo de trabajo.
- Debe considerar el manejo de errores. Por ejemplo en caso que no exista conexión a internet para consultar los datos o la validación de la dirección de correo electrónico.
- Debe descargar la imagen que indica el documento XML en un directorio local de su aplicación, por ejemplo: images
- Debe indicar el nombre y dirección URL de las librerías que utilizó a fin de configurar adecuadamente el ambiente de pruebas para su revisión.
- El tamaño de la ventana principal de su solución computacional debe ser de 800 x 600 píxeles, donde puede colocar un máximo de 3 películas por fila.
- En caso que el archivo start.xml no exista, debe crearlo en el directorio local.
- Puede considerar el uso de otro servicio de información de películas, entre tanto pueda extraer la información que se indicó en las imágenes adjuntas.
- En “Acerca de...” deben consignar la información de los integrantes del equipo.

Anexos

LLamada al OMDb API

Un ejemplo de como solicitar información al sitio de OMDb API es:

<http://www.omdbapi.com/?t=batman&y=&plot=full&r=xml>

Donde:

- t = corresponde al nombre de la película que se desea consultar
- plot = full solicita obtener la información completa
- r=xml correspondiente a solicitar la información de la misma vía XML

La respuesta a la petición anterior es como sigue:

```
<?xml version="1.0" response="True">
  <movie title="Batman" year="1989" rated="PG-13" released="23 Jun 1989"
runtime="126 min" genre="Action, Adventure" director="Tim Burton" writer="Bob Kane
(Batman characters), Sam Hamm (story), Sam Hamm (screenplay), Warren Skaaren
(screenplay)" actors="Michael Keaton, Jack Nicholson, Kim Basinger, Robert Wuhl"
plot="Gotham City. Crime boss Carl Grissom (Jack Palance) effectively runs the
town but there's a new crime fighter in town - Batman (Michael Keaton). Grissom's
right-hand man is Jack Napier (Jack Nicholson), a brutal man who is not entirely
sane... After falling out between the two Grissom has Napier set up with the
Police and Napier falls to his apparent death in a vat of chemicals. However, he
soon reappears as The Joker and starts a reign of terror in Gotham City.
Meanwhile, reporter Vicki Vale (Kim Basinger) is in the city to do an article on
Batman. She soon starts a relationship with Batman's everyday persona, billionaire
Bruce Wayne." language="French, English" country="USA, UK" awards="Won 1 Oscar.
Another 9 wins & 22 nominations." poster="http://ia.media-
imdb.com/images/M/MV5BMTYwNjAyODIyMF5BMl5BanBnXkFtZTYwNDMwMDk2._V1_SX300.jpg"
metascore="69" imdbRating="7.6" imdbVotes="263,082" imdbID="tt0096895"
type="movie"/>
</root>
```

Puntos a ser evaluados:

1. Correctitud de la solución computacional - 65%

Funcionalidad	Procesos	Valor
Carga del archivo XML	<ul style="list-style-type: none"> ● Abrir el archivo start.xml desde el directorio local. ● Parsear su contenido para formar los objetos de tipo Película, Actor y los que considere necesarios para implementar la tarea. ● Construir una lista en memoria que contiene objetos Película. 	20%
Ordenamiento alfabético	<ul style="list-style-type: none"> ● Algoritmo de ordenamiento alfabético de acuerdo al nombre de la película [A - Z]. Si el nombre de la película inicia con un valor numérico, debería colocarse de último. 	5%
Interfaz gráfica	<ul style="list-style-type: none"> ● Listado de películas ● Información de una película 	20% 15%
Envío de correo electrónico	<ul style="list-style-type: none"> ● Construir un mensaje de correo que contenga la información de la película que el usuario desea compartir. ● Validación de la dirección de correo del destinatario mediante expresión regular. 	10%
Solicitud http en línea	<ul style="list-style-type: none"> ● Consultar la información de acuerdo a los parámetros indicados anteriormente. ● Actualizar la información de la película en memoria. 	15%
Ranquear película	<ul style="list-style-type: none"> ● Debe dar persistencia a la puntuación que asigna el usuario mediante el XML. 	5%
Filtrar listado de películas	<ul style="list-style-type: none"> ● Reflejar la lista de películas que se despliega en pantalla. 	10%

2. Robustez de la solución computacional (validaciones) - 10%
3. Evitar los síntomas de un diseño pobre “olores del software” - 5%
 - a. Rigidez
 - b. Fragilidad
 - c. Inmovilidad
 - d. Viscosidad
 - e. Complejidad innecesaria
 - f. Repetición innecesaria
 - g. Opacidad
4. Entregar un documento con al menos los siguientes apartados: - 20%
 - a. Manual de usuario - 25 puntos
 - b. Pruebas de funcionalidad - 25 puntos

Debe demostrar evidencia de todas las funcionalidades implementadas en la tarea programada. Para esto realice la ejecución de su tarea programada y tome **screenshots** dejando evidencia del resultado de la ejecución de cada transacción.
 - c. Lecciones aprendidas - 15 puntos

Debe hacer un listado de todas las lecciones aprendidas producto del desarrollo de la tarea programada. Las lecciones aprendidas pueden ser de carácter personal y/o técnico.
 - d. Olores del software - 25 puntos

Debe indicar cuál o cuáles estrategias se usaron para apartarse de cada uno de los olores del software vistos en clase. Las estrategias deben sustentarse con segmentos de código de la tarea programada (tome screenshots) o bien con la documentación de las decisiones que fueron tomadas durante el diseño y la implementación.
 - e. Minutas y evidencias de asignación de responsabilidades - 10 puntos

Debe indicar las fechas y tiempos de cada una de las reuniones que realiza el equipo así como donde se realizan. También los temas tratados, las responsabilidades que son asignadas a los miembros del equipo y el seguimiento de los retos encontrados. Incluir un posible cronograma de cómo distribuir el trabajo a lo largo del tiempo asignado.
5. Debe entregar un CD/DVD, que contenga únicamente 2 carpetas llamadas: documentación y solución computacional, en la primera deberá incluir el documento de Word (no pdf) solicitado y en la segunda los archivos y/o carpetas necesarias para la implementación de esta tarea.

6. Entregue el CD/DVD en un sobre de manila sellado con sus calidades. La entrega es estrictamente personal en el momento que el profesor lo haya dispuesto.

a. Las calidades deben incluir:

- i. Nombre del curso
- ii. Número de semestre y año lectivo
- iii. Nombre del Estudiante
- iv. Número de carnet
- v. Número de tarea programada
- vi. Fecha de entrega
- vii. Estatus de la entrega (definido por el responsable de la

implementación de la tarea): [Deplorable|Regular|Buena|Muy Buena|Excelente|Superior]

7. La tarea será revisada en la versión de Python 3.5.1 únicamente.

Condiciones generales:

Esta tarea programada se rige por las siguientes condiciones:

Nota: El incumplimiento de alguna condición implicará una calificación de cero.

1. El desarrollo de la tarea es estrictamente en PAREJAS
2. La tarea DEBE implementarse con interfaz gráfica.
3. Debe cumplir con todo lo indicado en la sección "Puntos a ser evaluados"
4. Deberá entregarse en tiempo y forma según el plazo establecido por el profesor al momento la lectura de este documento.
5. El lenguaje de programación a utilizar es Python v3.5.1
6. Debe combinar la programación recursiva e iterativa para dar solución a esta tarea.
7. El día de entrega de la tarea corresponde al viernes 17 de junio de 2016.

Anexo 1: Manual de Usuario

1. Portada
2. Introducción
3. Qué funcionalidades implementadas tiene el software (estado general de la tarea)
4. Explicación paso a paso de cómo probar cada uno de los algoritmos implementados, esta explicación debe incluir el uso de casos de pruebas.
5. Pendientes de implementar y su justificación.
6. Bibliografía y Fuentes Digitales utilizadas para su investigación