

## Tarea Individual 18 - Constructores url

### Objetivo

Crear una aplicación en Java que haga uso de la clase `URL` para manipular y visualizar diferentes componentes de una URL. El objetivo es que el estudiante aprenda a crear instancias de la clase `URL` utilizando distintos constructores y a acceder a propiedades de la URL como el protocolo, el host, el puerto y otros detalles relevantes.

### Instrucciones

1. Crear un programa en Java que realice lo siguiente:
  1. Usar el constructor simple de la clase `URL` para crear una URL y visualizar sus propiedades. Ejemplo:
    - <http://docs.oracle.com>
  2. Utilizar otro constructor simple con una URL que contenga un directorio y parámetros de consulta. Ejemplo:
    - [http://localhost/PFC/gest/cli\\_gestion.php?S=3](http://localhost/PFC/gest/cli_gestion.php?S=3)
  3. Crear una URL utilizando el protocolo, el host y el directorio. Ejemplo:
    - <http://docs.oracle.com/javase/10/>
  4. Utilizar un constructor para crear una URL con protocolo, host, puerto y directorio. Ejemplo:
    - <http://localhost:8084/WebApp/Controlador?accion=modificar>
  5. Utilizar el constructor con contexto de otra URL para crear una URL completa a partir de una base. Ejemplo:
    - <https://docs.oracle.com/javase/10/docs/api/java/net/URL.html>

```
// 1. Constructor simple: URL completa
URL url1 = new URL("http://docs.oracle.com/");
imprimirPropiedadesURL("URL 1: Constructor simple", url1);

// 2. Constructor simple con directorio y parámetros de consulta
URL url2 = new URL("http://localhost/PFC/gest/cli_gestion.php?S=3");
imprimirPropiedadesURL("URL 2: Con directorio y parámetros", url2);

// 3. Constructor con protocolo, host y directorio
URL url3 = new URL("http", "docs.oracle.com", "/javase/10/");
imprimirPropiedadesURL("URL 3: Protocolo, host y directorio", url3);

// 4. Constructor con protocolo, host, puerto y directorio
URL url4 = new URL("http", "localhost", 8084, "/WebApp/Controlador?accion=modificar");
imprimirPropiedadesURL("URL 4: Protocolo, host, puerto y directorio", url4);

// 5. Constructor con contexto de otra URL (ruta relativa)
URL baseURL = new URL("https://docs.oracle.com/javase/10/");
URL url5 = new URL(baseURL, "docs/api/java/net/URL.html");
imprimirPropiedadesURL("URL 5: Con base y ruta relativa", url5);
```

2. Visualizar las propiedades de cada URL creada, incluyendo:
  - El protocolo (`getProtocol()`)
  - El host (`getHost()`)
  - El puerto (`getPort()`)
  - El archivo (`getFile()`)
  - El usuario (`getUserInfo()`)

- El path (`getPath()`)
- La consulta (`getQuery()`)
- La autoridad (`getAuthority()`)
- El puerto por defecto (`getDefaultPort()`)

```
URL 1: Constructor simple
Protocolo: http
Host: docs.oracle.com
Puerto: -1
Archivo: /
Usuario: null
Path: /
Consulta: null
Autoridad: docs.oracle.com
Puerto por defecto: 80

URL 2: Con directorio y parámetros
Protocolo: http
Host: localhost
Puerto: -1
Archivo: /PFC/gest/cli_gestion.php?S=3
Usuario: null
Path: /PFC/gest/cli_gestion.php
Consulta: S=3
Autoridad: localhost
Puerto por defecto: 80

URL 3: Protocolo, host y directorio
Protocolo: http
Host: docs.oracle.com
Puerto: -1
Archivo: /javase/10/
Usuario: null
Path: /javase/10/
Consulta: null
Autoridad: docs.oracle.com
Puerto por defecto: 80

URL 4: Protocolo, host, puerto y directorio
Protocolo: http
Host: localhost
Puerto: 8084
Archivo: /WebApp/Controlador?accion=modificar
Usuario: null
Path: /WebApp/Controlador
Consulta: accion=modificar
Autoridad: localhost:8084
Puerto por defecto: 80

URL 5: Con base y ruta relativa
Protocolo: https
Host: docs.oracle.com
Puerto: -1
Archivo: /javase/10/docs/api/java/net/URL.html
Usuario: null
Path: /javase/10/docs/api/java/net/URL.html
Consulta: null
Autoridad: docs.oracle.com
Puerto por defecto: 443
```

3. Para realizar pruebas con un proxy, configura las propiedades de `http.proxyHost` y `http.proxyPort` (aunque esta parte puede estar comentada en el código).

```
// Configuración de proxy (opcional y comentada para pruebas)
System.setProperty("http.proxyHost", "192.168.1.30");
System.setProperty("http.proxyPort", "3128");
```

```
// Limpiar el proxy después de las pruebas
System.clearProperty("http.proxyHost");
System.clearProperty("http.proxyPort");
```

## Consideraciones

Asegúrate de manejar correctamente las excepciones como `MalformedURLException` y de utilizar correctamente los métodos de la clase `URL` para acceder a la información deseada.

```
} catch (MalformedURLException e) {  
    System.err.println("Se ha producido un error al construir una URL: " + e.getMessage());  
}
```