

# Programación de Aplicaciones

## Tarea 3

### Curso 2019

## 1 Información Administrativa

La tarea comienza el **lunes 21 de octubre** y finaliza el **domingo 17 de noviembre**. La tarea constará de múltiples entregas parciales a realizarse durante las clases de monitoreo, y la entrega final de código a realizarse hasta el **domingo 17 de noviembre** a las **23:59hs** (ver [1]).

## 2 Objetivos

La presente tarea pretende la puesta en práctica y profundización por parte del estudiante de los siguientes puntos:

- Lograr la interoperabilidad de aplicaciones distribuidas vía *Web Services*
- Aplicar técnicas básicas de verificación de aplicaciones
- Desplegar las aplicaciones en entornos de producción
- Aplicar prácticas usuales en el desarrollo de software
  - o Trabajo en grupo
  - o Planificación de actividades
  - o Registro de horas

## 3 Introducción

Basados en los lineamientos definidos en la Descripción del Proyecto [2], en esta tercera etapa, se continuará con el desarrollo de los componentes Servidor Central y Servidor Web y se desarrollará el componente Dispositivo Móvil. Al finalizar esta iteración se contará con las aplicaciones ejecutando cada una de ellas de forma independiente en sus nodos correspondientes, comunicándose entre sí.

Si bien existen diversas alternativas para el desarrollo del Dispositivo Móvil, se optó por desarrollar una aplicación web con ciertas particularidades, como se presentó en la arquitectura de la descripción general del proyecto que se observa en la Figura 1.

A la hora de considerar el mecanismo de comunicación y teniendo en cuenta la amplia gama de dispositivos en el mercado, se valora especialmente la interoperabilidad del sistema. El intercambio de datos mediante Web Services [3] contempla esta situación ya que se trata de un protocolo de comunicación basado en estándares.

En esta tarea el Servidor Central es quién se encarga de centralizar los datos de la ejecución. La lógica del negocio que provee el Servidor Central se invoca en forma directa desde la Estación de Trabajo, y a la vez, se expone como Web Services para ser accedida en forma remota desde el Servidor Web, para atender tanto al cliente web como al dispositivo móvil.

En conclusión, se deberán distribuir los componentes de acuerdo a la arquitectura planteada, como se muestra en la Figura 1, mediante el uso de Servicios Web, entre el Servidor Web y el Servidor Central.

**Nota:** El Dispositivo Móvil **NO** se debe desarrollar con la J2ME (Java 2 Platform, Micro Edition).

**Figura 1: Diagrama de Distribución de la 3er. iteración**

## 4 Visión

En esta iteración se potenciarán las funcionalidades de los distintos tipos de actores. Adicionalmente, se incorporará un nuevo componente que permitirá a los colaboradores acceder a la plataforma desde un Dispositivo Móvil extendiendo las posibilidades de uso del sistema. A continuación se mencionan algunas de las funcionalidades agregadas.

### Dispositivo Móvil

Debido a que hoy en día es cada vez más popular acceder a los sitios a través de dispositivos como tabletas, teléfonos inteligentes, libros electrónicos, etc; es necesario que las interfaces sean adaptables. Por tal motivo, el dispositivo móvil se desarrollará como una aplicación web siguiendo los lineamientos de "Responsive Web Design" (RWD), esto es, un solo diseño web cuya visualización se adecue a cualquier dispositivo.

### Experiencia de Usuario

En base a la información sobre las consultas a videos, se brindará una lista de históricos que será ordenada en base a las consultas favoritas que el usuario realice. Además, se permitirá compartir de una manera sencilla las direcciones de los videos, lo que permitirá generar mayor interacción entre usuarios y fidelidad para con la plataforma.

## 5 Actores

Los actores en esta etapa son:

- **Administrador del Sistema:** utiliza solamente la estación de trabajo, esto implica que no puede iniciar sesión en el sitio Web.
- **Visitante:** es cualquier usuario que se encuentra en la Web pero no inició sesión como usuario.
- **Propietario:** un usuario que inició sesión como propietario de un canal y no la cerró. Entre otras funcionalidades, creará videos y listas de reproducción.

La distinción entre Visitante y Propietario es para indicar qué Casos de Uso pueden ejecutar.

## 6 Requerimientos Funcionales

Para esta iteración del proceso de desarrollo se han seleccionado para su implementación, o modificación, los siguientes casos de uso.

### 6.1 Estación de Trabajo

Caso de uso	Ver Usuarios Eliminados
Actores	Administrador del Sistema
Descripción	Este caso de uso permite al Administrador ver los datos básicos de los Usuarios eliminados, su canal y los contenidos (videos y listas) del mismo, así como la fecha de eliminación. Una parte de estos datos será (opcionalmente) recuperados del almacenamiento persistente en disco como se indica en el <b>Requerimiento Especial 7.2 Persistencia de datos de Usuario eliminado</b> .

### 6.2 Servidor Web

Caso de uso	Baja de Usuario
Actores	Propietario
Descripción	Este caso de uso permite a un propietario darse de baja de la plataforma. Para ello, el propietario selecciona la opción para darse de baja y el sistema elimina toda la información relacionada con éste, como ser: datos básicos, imágenes, canal, videos y listas de reproducción propias, videos en listas de reproducción de otros usuarios, las relaciones del tipo “seguir” existentes entre el propietario y otros usuarios del sistema, las valoraciones y comentarios que el usuario haya realizado, así como las valoraciones y comentarios que otros usuarios hayan realizado a sus videos. Una parte importante de estos datos, será (opcionalmente) persistida en disco, tal como se explica en el Requerimiento Especial 7.2 Persistencia de Datos de Usuario Eliminado.

### 6.3 Dispositivo Móvil

Para el diseño de este componente se deberá tener en cuenta el **Requerimiento Especial 7.1 Diseño del Dispositivo Móvil**.

Nombre	Inicio de Sesión
Actores	Visitante
Sinopsis	Este caso de uso permite a un visitante iniciar sesión mediante el ingreso de su nick o mail y contraseña. El sistema verifica que los datos sean válidos e inicia la sesión para el propietario.
Nombre	Cierre de Sesión
Actores	Propietario
Sinopsis	Este caso de uso permite cerrar la sesión iniciada por un propietario. Para ello, el usuario debe haber iniciado sesión previamente.
Nombre	Consulta de Video

Actores	Propietario
Sinopsis	El caso de uso comienza cuando un usuario desea consultar un video. En primer lugar, el usuario ingresa un término asociado al título de un video que desea consultar. El sistema lista los videos existentes cuyo nombre o descripción contengan el término ingresado. El usuario elige uno de ellos y el sistema muestra los datos básicos del video y opcionalmente los comentarios del mismo (incluyendo autor y fecha del comentario) y cantidad de valoraciones de los usuarios (a cuántos les gusta y a cuántos no).
Nombre	Consulta de Lista de Reproducción
Actores	Propietario
Sinopsis	El caso de uso comienza cuando un usuario quiere consultar la información de una de sus listas de reproducción. En primer lugar, el sistema muestra las listas de reproducción (propietarias y por defecto, públicas y privadas) del usuario propietario. El usuario selecciona una lista y el sistema devuelve los datos de la lista de reproducción: nombre, categorías y lista de videos en ella. En caso de seleccionar un video, se puede ver su información como indica el caso de uso Consulta de Video.

## 7 Requerimientos Especiales

### 7.1 Diseño del Dispositivo Móvil

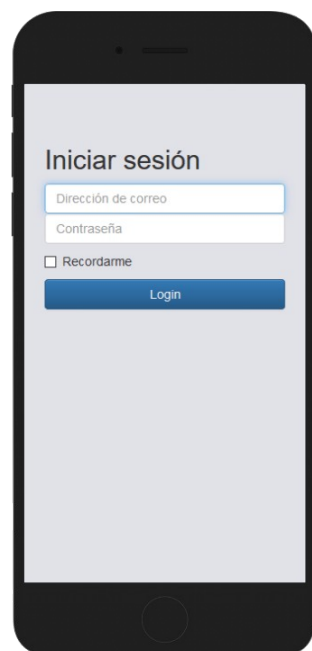
El desarrollo del Dispositivo Móvil como una aplicación web para los clientes añade algunos requerimientos al diseño de la interfaz. Debido a que hoy en día es cada vez más popular acceder a los sitios a través de dispositivos como tabletas, teléfonos inteligentes, libros electrónicos, etc; es necesario que las interfaces sean adaptables. En consecuencia se seguirán los lineamientos del paradigma "Responsive Web Design" (RWD): un solo diseño web que se adecue a cualquier dispositivo.

Esta aplicación web será independiente del Sitio Web de UyTube ya que tendrá distinto diseño y funcionalidades. Aun así, no es necesario que sean dos aplicaciones diferentes, es decir, se pueden ubicar en el mismo servidor web, e incluso ser accesibles desde la misma URL.

En cuanto al diseño de la aplicación, se utilizará Bootstrap [4] o algún otro framework para RWD y se deberán contemplar los siguientes aspectos.

#### Inicio de Sesión

La página inicial será la de *login*, correspondiente al caso de uso *Inicio de Sesión*, donde el propietario deberá ingresar su usuario y contraseña, con la posibilidad de recordarlo. Para ello se deberá considerar el diseño presente en la Figura 2 (ver *Signin* [5]).

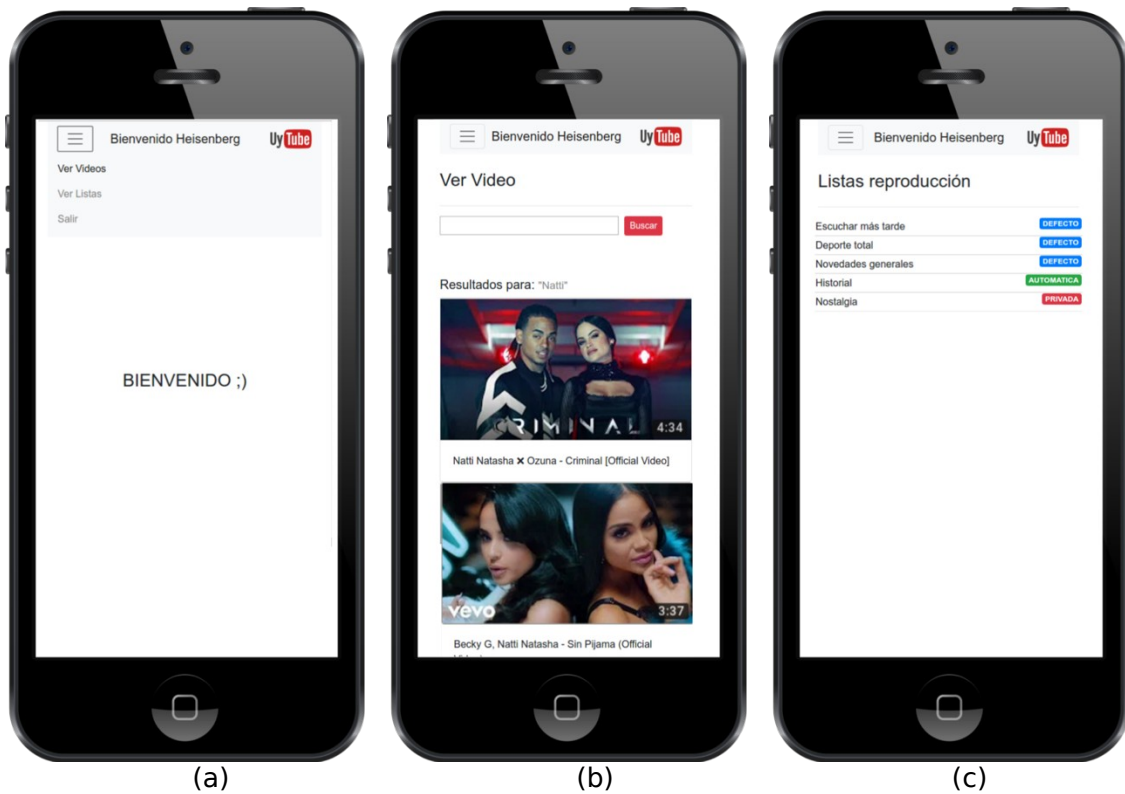


**Figura 2: Diseño de la página de login de un usuario**

## **Barra de Navegación**

Una barra de navegación es un cabezal ubicado en lo más alto del sitio web que permite navegar a diferentes lugares del sitio, una vez el usuario se encuentra autenticado, como se muestra en la Figura 3(a) (ver *NavBar* [6]).

En particular, es deseable contar con una que permita acceder a las páginas para realizar los casos de uso Consulta de Video, Consulta de Lista de Reproducción y Cierre de Sesión. Esta barra de navegación tendrá que adaptarse al realizar modificaciones en las dimensiones de la pantalla, permitiendo acceder a las diferentes secciones a través de un ícono "hamburger", tal como se indica en la Figura 3(a).



**Figura 3: Diseño de la barra de navegación, consulta de videos y listas**

## Listado de Elementos

Para realizar los casos de uso *Consulta de Álbum* y *Consulta de Lista de Reproducción*, es necesario mostrar elementos como videos y listas de reproducción debajo de la barra de navegación y ajustando su visualización acorde a la redimensión de la pantalla. Una forma de hacer esto es mediante una grilla (ver *Grid* [7]) que contenga una foto del video/lista de reproducción con su nombre, como se muestra en la Figura 3(b) que permita al hacer click sobre ella ir a la vista que muestra la información completa del video/lista de reproducción.

Por su parte, para visualizar los videos asociados a una lista de reproducción, se pueden utilizar tablas (ver *Tablas*[8]), tal como se muestra en la Figura 3(c).

## Reproducción de temas (opcional)

Al igual que la aplicación web, la posibilidad de mirar videos es fundamental para una plataforma como UyTube. Por esta razón, interesa que la aplicación móvil cuente con un reproductor de videos similar al que se tiene en la aplicación web.

El reproductor deberá encontrarse en la parte inferior de la pantalla en todo momento y ajustar su visualización acorde a la redimensión de la pantalla.

## 7.2 Calidad del Producto

La definición y ejecución de planes de pruebas adecuados es clave en el desarrollo de un sistema de alta calidad. En este sentido, al igual que lo solicitado en la iteración anterior, se requerirá el mantenimiento y desarrollo de pruebas automáticas para el componente Servidor Central. Estas pruebas deberán ser automatizadas utilizando JUnit [8] y la evaluación del cubrimiento de las mismas deberá realizarse con EcEmma [9]. Se requiere como mínimo que el cubrimiento de sentencias alcance el 80% del código que corresponde a la capa lógica. Cabe aclarar que solamente interesa la realización de pruebas sobre el componente

Servidor Central, es decir, no es la intención que se realicen pruebas automáticas de interfaz de usuario ni sobre el componente Servidor Web.

La calidad no depende exclusivamente de la ausencia de errores en el código, sino de otros factores, por ejemplo la calidad del código generado. Cada organización suele definir buenas prácticas y exigir estándares de codificación que permiten, entre otras cosas, obtener un código más legible que facilite la realización de pruebas y el mantenimiento del código. En esta tarea se integrarán diversas herramientas de calidad del software como plugin de Eclipse, en particular las herramientas PMD [10] y CheckStyle [11] que permiten realizar un análisis estático del código (sin ejecutarlo) para identificar diversos problemas y comprobar el cumplimiento de buenas prácticas. En ambos casos se deberá modificar el código para que no aparezcan errores en ninguna de las siguientes categorías de evaluación. Se deberá usar la configuración básica, sin modificar, e incluir los chequeos que sean necesarios si no se encuentran habilitados.

<b>PMD</b>	<b>CheckStyle</b>
AbstractClassWithoutAbstractMethod LooseCoupling PackageCase ShortVariable UnusedPrivateField	AvoidStarImport EmptyCatchBlock GenericWhitespace IllegalCatch LeftCurly MemberName MethodName MethodParamPad UnnecessaryParentheses UnusedImports VisibilityModifier WhitespaceAfter

### 7.3 Favoritos Históricos

UyTube ofrecerá a los usuarios una lista de reproducción por defecto de nombre “Historial” que será generada automáticamente con aquellos videos que el usuario haya consultado. La lista tendrá los videos ordenados según la cantidad de veces que dicho usuario haya consultado el video (estando con su sesión iniciada). Esta información (cantidad de consultas realizadas para cada video) será indicada al consultar la información de la lista, junto con la última fecha de consulta de cada video.

### 7.4 Compartir Video

Al momento de realizar la Consulta de Video, el usuario tendrá la opción de compartir un video. Esta funcionalidad generará una dirección URL acortada <http://uytube.com/COD> que hará referencia a la página de consulta de información del video dentro de la plataforma UyTube, la cual podrá ser compartida por el usuario. Para ello es necesario generar un identificador único para cada video cargado en la plataforma (CODIGO). En caso de que un usuario ingrese la URL en su explorador, la aplicación deberá detectar que se desea acceder a un video en base a su código y redireccionará la petición a la página correspondiente de consulta de dicho video. Para ello, es necesario utilizar filtros, tal como se explica en [12].

### 7.5 Validación de Datos

El ingreso de datos por parte del usuario vía Web deberá ser validado a nivel del cliente (navegador), ya sea exigiendo que las entradas se seleccionen desde un combo box o validando los datos mediante JavaScript [13]. Validar los datos

significa, en este caso, verificar el tipo y dominio de los datos, la buena formación de los formularios (ingreso de campos obligatorios, etc.), en particular en el caso de uso Alta de Usuario donde se tiene que verificar que la contraseña y su confirmación coincidan.

Además, en el caso de uso Alta de Usuario se deberá verificar automáticamente la disponibilidad del nickname y del email antes de que el usuario envíe sus datos por el formulario web. Este tipo de verificación se deberá realizar a través de AJAX [14] en segundo plano a medida que el usuario escribe el nickname o el email en los cuadros de texto. La interfaz web deberá indicar si el nickname o email está disponible o no con un mensaje junto al cuadro de texto. La Figura 5 muestra un ejemplo de interfaz para el caso en que haya un usuario con el nickname “kairoh”, pero el nickname “kairoh2” esté disponible.

**Figura 4: Control de disponibilidad de nick y sugerencia asociada**

## 8 Se Pide

- Documentación: Análisis y diseño de los aspectos más relevantes de la realidad
- Gestión: Planificación del trabajo grupal y registro de horas de actividad de cada integrante del grupo
- Desarrollo:
  - o Desarrollo de una interfaz gráfica de usuario Web para el dispositivo móvil.
  - o Desarrollo y modificación de los componentes de la arquitectura para cumplir con los nuevos requerimientos.
  - o Desarrollo de los Requerimientos Especiales.

En la sección Laboratorio link "Calendario de entregas y evaluaciones" del sitio EVA del curso se encuentra la indicación de entregas intermedias, final y sus fechas asociadas, así como los requerimientos mínimos y no mínimos para su aprobación.



## 9 Referencias

[1]	Estándar de Entregas <a href="https://eva.fing.edu.uy/course/view.php?id=315">https://eva.fing.edu.uy/course/view.php?id=315</a>
[2]	Descripción del Proyecto <a href="http://eva.fing.edu.uy/course/view.php?id=315">http://eva.fing.edu.uy/course/view.php?id=315</a>
[3]	Metro Web Services Overview <a href="http://www.oracle.com/technetwork/java/index-jsp-137004.html">http://www.oracle.com/technetwork/java/index-jsp-137004.html</a>
[4]	Bootstrap <a href="http://getbootstrap.com">http://getbootstrap.com</a>
[5]	Bootstrap-Signin <a href="http://getbootstrap.com/examples/signin">http://getbootstrap.com/examples/signin</a>
[6]	Bootstrap- Navbar <a href="http://getbootstrap.com/examples/navbar-fixed-top">http://getbootstrap.com/examples/navbar-fixed-top</a>
[7]	Bootstrap- Grid <a href="http://getbootstrap.com/examples/grid">http://getbootstrap.com/examples/grid</a>
[8]	Bootstrap- Tables <a href="http://www.w3schools.com/bootstrap/bootstrap_tables.asp">http://www.w3schools.com/bootstrap/bootstrap_tables.asp</a>
[9]	HSQLDB - HyperSQL DataBase <a href="http://hsqldb.org/">http://hsqldb.org/</a>
[10]	H2 Database Engine <a href="http://www.h2database.com/html/main.html">http://www.h2database.com/html/main.html</a>
[11]	JPA - Java Persistence API <a href="http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html">http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html</a>
[12]	The Essentials of Filters <a href="http://www.oracle.com/technetwork/java/filters-137243.html">http://www.oracle.com/technetwork/java/filters-137243.html</a>
[13]	JavaScript <a href="https://developer.mozilla.org/es/JavaScript">https://developer.mozilla.org/es/JavaScript</a>
[14]	AJAX Introduction <a href="https://www.w3schools.com/js/js_ajax_intro.asp">https://www.w3schools.com/js/js_ajax_intro.asp</a>
[15]	Tomcat Application Developer's Guide - Deployment <a href="http://tomcat.apache.org/tomcat-7.0-doc/appdev/deployment.html">http://tomcat.apache.org/tomcat-7.0-doc/appdev/deployment.html</a>
[16]	Sample web.xml file. <a href="http://tomcat.apache.org/tomcat-7.0-doc/appdev/web.xml.txt">http://tomcat.apache.org/tomcat-7.0-doc/appdev/web.xml.txt</a>
[17]	JAX-WS Hello World Example - RPC Style <a href="http://www.mkyong.com/webservices/jax-ws/jax-ws-hello-world-example/">http://www.mkyong.com/webservices/jax-ws/jax-ws-hello-world-example/</a>
[18]	The Java Tutorials - Properties. <a href="http://download.oracle.com/javase/tutorial/essential/environment/properties.html">http://download.oracle.com/javase/tutorial/essential/environment/properties.html</a>
[19]	Apache Ant <a href="https://ant.apache.org/">https://ant.apache.org/</a>
[20]	Apache Maven <a href="https://maven.apache.org/">https://maven.apache.org/</a>