



1^{ER} PROYECTO 20%

Fecha de entrega: 01/07/2015

Objetivo:

Desarrollar habilidades en programación de aplicaciones comerciales con Software Libre, mediante la creación de una aplicación Web y de Línea de Comandos, que se integre con aplicaciones/librerías de terceros, para el reforzamiento de los conocimientos adquiridos en clase.

Instrucciones generales:

- La realización del proyecto es en parejas.
- Es prohibido copiar trabajos, cualquier intento de fraude o plagio será castigado todo el peso del reglamento institucional.
- La defensa del proyecto tendrá lugar el día 01 de Julio del 2015, en la UTN, sede San Carlos (misma aula donde se recibe el curso regular, en el horario habitual).
- Deben distribuir el tiempo sabiamente, para que al final no se les acumule el trabajo y terminen dejando puntos importantes sin cumplir (ver la tabla de rubros por calificar).

Enunciado

El restaurante temático **“El Programador Feliz”** lo ha contratado a usted para que desarrolle un sistema inteligente para la gestión del ambiente musical del lugar. El conjunto de software que se debe desarrollar deberá cumplir con los siguientes requerimientos:

Aplicación Web:

Se deberá construir una aplicación web, que deberá contar con dos interfaces, una administrativa que permitirá gestionar el sistema y otra de usuario final que funcionará como la interfaz de usuario de una “Cyber Rockola”.

- **La interfaz administrativa:**

Permitirá a los usuarios con rol de “administrador” ejecutar tareas como:

- Iniciar sesión en el sitio (sólo si se autentica con credenciales válidas)
- Registrar y editar artistas (cantantes)
- Crud de canciones (se debe incluir mínimo, el archivo, el nombre de la

cancion y el nombre del artista)

- Encolar canciones en la “Cyber Rockola” (Se debe encolar un mensaje en algún servidor de colas, se sugiere el uso de “Redis” o “RabbitMQ”)

- **La interfaz de usuario final:**

Permitirá a los usuarios con rol de “cliente” ejecutar tareas como:

- Registrarse como usuario en la “Cyber Rockola”
- Iniciar sesión en el sitio (sólo si se autentica con credenciales válidas)
- Ver lista de canciones disponibles
 - Se deberá utilizar paginación para mostrar las canciones en grupos de 15 elementos por página
 - Se deberá implementar un mecanismo de filtro por artista y por titulo
- Seleccionar y encolar canciones en la “Cyber Rockola”. Se debe encolar un mensaje en algún servidor de colas, se sugiere el uso de “Redis” o “RabbitMQ”.

Módulo CLI - Worker:

Se requiere desarrollar una aplicación de línea de comandos que hará la función de “listener” de la cola y se encargará auto-gestionar la reproducción de las canciones.

- La aplicación de consola debe monitorear el servidor de colas y cuando entren mensajes (canciones en cola), debe iniciar la reproducción lo más rápido posible.
- Si hay más de una canción en la cola, el “worker” debe esperar a que termine la reproducción de la canción que está sonando, para continuar con la siguiente.
- Las canciones se deben ir reproduciendo respetando el orden en que ingresaron a la cola (FIFO).

Otros requerimientos técnicos

Además de desarrollar la “Aplicación Web” con sus dos interfaces y el “Módulo CLI – Worker”, el sistema deberá satisfacer los siguientes requerimientos técnicos:

- La “Aplicación Web” no debe reproducir la música, la reproducción tendrá lugar el sistema operativo anfitrión del “Módulo CLI – Worker”.
- La cola de canciones debe ser estrictamente implementada con un servidor de colas, se sugieren “RabbitMQ” o “Redis”. No se debe guardar información relacionada a la cola, en base de datos.
- Las rutas del proyecto deben ser las siguientes:

VERBO HTTP	URL	Descripción
GET	/songs	Muestra el listado de canciones
POST	/songs	Crear una canción
PUT o PATCH	/songs/{id}	Edita una canción
DELETE	/songs/{id}	Elimina una canción
GET	/songs/create	Muestra el formulario para subir una nueva canción
GET	/songs/{id}/edit	Muestra el formulario para editar una canción existente
GET	/songs?name=nombre	Filtro por nombre
GET	/songs?artist=nombre	Filtro por artista
GET	/songs?artist=nombre&name=nombre	Filtro por nombre y artista
POST	/songs/{id}/enqueue	Agrega una canción X a la cola de reproducción

- En base de datos solamente tendrá lugar el almacenamiento de la información relacionada con los artistas, las cuentas de usuario y los datos de las canciones disponibles como lo son: el título, el artista y la ubicación física del archivo de audio.
- La “Aplicación Web” soportará dos tipos de roles de usuario, “Administrador” y “Cliente”. A los usuarios que se registran de manera autónoma en el sitio, se le auto-asignará el rol de “Cliente”.
- Solamente los usuarios con el rol de “Administrador” podrán ingresar a la “Interfaz Administrativa”. Al iniciar la aplicación por primera vez debe existir un usuario “Built-in” con el rol “Administrador”.
- Se debe utilizar algún “UI framework” para la interfaz gráfica.
- Se debe utilizar ya sea MySQL o PostgreSQL como motor de base de datos y de Git como control de versiones (subir repositorio a BitBucket o Gitlab).

Rubros de Calificación

Esta asignación tiene un valor de 20% de la nota total del curso. El detalle de cada uno de los rubros a evaluar se presenta a continuación.

1 ^{er} Proyecto Programado	Puntos
Dominio del tema	
Explica el funcionamiento del código	5
Responde a las preguntas acertadamente	5
La Aplicación Web	
Implementa mecanismo de “Login” para usuarios administrativos y clientes	2.5
CRUD de canciones	10
Permite al cliente registrarse como usuario en la “Cyber Rockola”	5
Permite visualizar la lista de canciones disponibles	5
Utiliza paginación para mostrar la lista de canciones disponibles	5
Implementa mecanismo de filtrado por artista y/o título de la canción	10
Permite seleccionar y encolar canciones en la “Cyber Rockola”	5
El Módulo CLI - Worker	
Implemente mecanismo para monitorear cola de canciones	5
Espera la finalización de la canción que está sonando	5
Reproduce las canciones en el orden en cual llegaron a la cola	5
Otros Requerimientos Técnicos	
Respetar el sistema de rutas requerido	10
Implementa la cola de canciones estrictamente con servidor de colas	10
Implementa correctamente la lógica de roles de usuario	5
Utiliza un “UI Framework” para interfaz gráfica de la “Aplicación Web”	5
Utiliza Git desde el principio como CVS y publica repositorio en Gitlab o Bitbucket	2.5

Referencias

Rabbit Technologies Ltd. (2015). Clients & Developer Tools. 08/06/2015, de RabbitMQ Sitio web: <http://www.rabbitmq.com/devtools.html>

Michael Hipp, Oliver Fromme. (2015). Fast console MPEG Audio Player and decoder library. 08/06/2015, de mpg123 Sitio web: <http://www.mpg123.de/>

Alex Beregszászi. (2015). MPlayer Official Site. 08/06/2015, de MPlayer Team Sitio web: <https://www.mplayerhq.hu>

Anexo

Instalación de RabbitMQ en OpenSuse:

```
$ sudo zypper install rabbitmq-server rabbitmq-server-plugins
```

Instalación de RabbitMQ en Ubuntu/Debian (desde repositorios de la distro):

```
$ sudo apt-get install rabbitmq-server
```

Instalación Ubuntu/Debian (desde repositorios de RabbitMQ):

<http://www.rabbitmq.com/install-debian.html>