



**Pontificia Universidad Católica Madre y Maestra**

**TEP**

**Asignado A:**

Richard Yunior Gómez Peña

**Asignado Por:**

Luis Roja

**Materia:**

Programación Web II

**Matricula:**

0014-0176

**2do Parcial**

Santiago de los caballeros, Rep. Dominicana

9/7/2018



- 1 – Cree mi proyecto gradle en el IDE Eclipse
- 2 – Descargue los Links requeridos por Spark Java y Gradle.

testImplementation 'junit:junit:4.12'

// <http://sparkjava.com/documentation#getting-started>

compile "com.sparkjava:spark-core:2.7.2"

// <https://mvnrepository.com/artifact/org.slf4j/slf4j-api>

compile group: 'org.slf4j', name: 'slf4j-api', version: '1.8.0-beta2'

// <https://mvnrepository.com/artifact/org.slf4j/slf4j-simple>

testCompile group: 'org.slf4j', name: 'slf4j-simple', version: '1.8.0-beta2'

- 3 – Cree las hojas siguientes:

- Index.html
- About.html
- Music.html
- Notfound.html

- 4 – Cree en mi carpeta resources los siguiente:

- Images.css
- Inconos.font
- Iconos.css
- Css

## **INTRODUCION:**

En este proyecto se basa a desarrollar mi primer proyecto con Eclipse

Utilizando las herramientas asignadas como Spark y Gradle.

En esta web veremos cómo inicializamos un Servidor con Gradle y los Frameworks de Spark Java.

En la web podemos apreciar los detalles como HTML, Css, y JavaScript.

### **Planteamiento de Problema:**

Una empresa discográfica ha dejado en sus manos un producto mínimo viable de una aplicación para nuevos artistas que estos pretenden lanzar. La descripción de la aplicación, según la empresa discográfica, se presenta a continuación:

“Requerimos una aplicación la cual permita a nuevos artistas registrarse y publicar sus discos en nuestro sitio web. Cada disco puede componerse de una o más canciones, y es necesario que estos se registren en la aplicación antes de poder subir un disco. De un disco, entendemos que es necesario el nombre del disco, nombre del artista, imagen del disco, la fecha de lanzamiento y el país donde se grabó tal disco. De las canciones, solo queremos, por el momento, saber el nombre y si se grabaron o no con otra persona. No obstante, no es necesario especificar a la otra persona. También, queremos que sea posible agregar reviews a los discos. De 1 a 5 estrellas, y, de ser posible (Aunque no completamente necesario), dejar un comentario.”.

## **Objetivo**

El objetivo en esta meta es desarrollar una página web con calidad, y correctas funcionalidades. El cual el cliente se sienta a gusto con lo dado.

## **Mi objetivo**

Es poner en desarrollo mi capacidad para esta materia como programador, destacar, aprender y tener nuevos objetivos siempre como el de este segundo parcial.

## Conclusión

El fin de esta web esta realizada para que los clientes que quieran visitar la web con fines de escuchar y descargar musica gratis.

En la misma podemos encontrar una hermosa página bien detallada con un menú responsivo, la misma está diseñada para ser ajustada a cualquier tipo de pantalla.

## Servidor

```
import static spark.Spark.*;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Servidor {

    public static void main(String[] args) {

        staticFileLocation("/");

        Logger logger = LoggerFactory.getLogger(Servidor.class);

        get("/index", (req, res) -> {
            logger.info("/index :: entramos por home");
            res.redirect("/index.html");

            return null;
        });

        get("/about", (req, res) -> {
            logger.info("/about :: entramos por about");
            res.redirect("/about.html");
```

```
    return null;
});
```

```
get("/index", (req, res) -> {
    res.redirect("/index.html");
    return null;
});
```

```
    get("/about", (req, res) -> {
        res.redirect("/about.html");
        return null;
```

```
});
```

```
get("/notfound", (req, res) -> {
    res.status(404);
    return "NOT FOUND";
});
```

```
}
```

```
}
```



```

before((req, res) -> {
    //Revisamos si la Cookie existe
    if(req.session(true).attribute("usuarioEnSesion") == null
        && req.cookie("usuarioLoggeado") != null) {

        String nombreDeUsuario = req.cookie("usuarioLoggeado");

        for(Usuario user : usuarios) {
            //Guardamos en la session si existe el nombre de usuario
            if(user.getUsuario().equals(nombreDeUsuario)) {
                req.session(true).attribute("usuarioEnSesion", user);
            }
        }
    }
});

//Protegemos la ruta, para quienes no esten registrado
before("/privado/*", (req, res) -> {
    if(req.session(true).attribute("usuarioEnSesion") == null) {
        halt(401, "Usted no esta loggeado");
    }
});

get("/", (req, res) -> {
    if(req.session(true).attribute("usuarioEnSesion") == null) {
        res.redirect("/formulario.html");
        return null;
    }
}

```

```

Usuario usuarioEnSesion = req.session(true).attribute("usuarioEnSesion");

    return "Bienvenido " + usuarioEnSesion.getNombre() + " " +
usuarioEnSesion.getApellido();
});

//Nos permite tener nombre de usuario y contraseña
post("/login", (req, res) -> {
    String nombreUsuario = req.queryParams("usuario");
    String contraseña = req.queryParams("pass");
    boolean rememberMe = req.queryParamOrDefault("rem",
"off").equals("on");

    for(Usuario user : usuarios) {
        if(user.getUsuario().equals(nombreUsuario)
            && user.getContraseña().equals(contraseña)) {
            req.session(true).attribute("usuarioEnSesion", user);
            if(rememberMe) {
                res.cookie("usuarioLoggeado", user.getUsuario());
            }
        }
    }
    res.redirect("/");
    return null;
});

get("/privado/este", (req, res) -> {
    return "So lo por usuarios loggeados ESTE";
});

get("/privado/tambien", (req, res) -> {
    return "Solo por usuarios loggeados TAMBIEN";
});

```

}

}