## Documentación para Proyecto Java Web Puro

### 1. Definición del Proyecto

* **Nombre del Proyecto: Cine-utn**
* **Objetivo:** es una aplicación web para comprar entradas de cine
* **Alcance:** CRUD de películas y funciones, compra de entradas para películas específicas (sin visualización de butacas).

### 2. Descripción General

* **Tecnología Utilizada:**
  + Lenguaje de programación: Java
  + Utilizamos Jakarta EE
  + Base de datos: utilizamos MySQL, utilizamos como interfaz de diseño MySQL workbench
  + Servidor de aplicaciones: utilizamos Tomcat 10
  + Utilizamos JDBC para interactuar con la base de datos
* **Arquitectura:**
  + Utilizamos la arquitectura MVC, modelo vista controlador, el modelo seria las clases beans, las clases DAO y la interacción con la base de datos, las vistas serian los jsp, y los controladores serian los servlets y filters
* **Flujo de Trabajo General:**
  + Descripción de cómo interactúan los diferentes componentes del sistema

### 3. Rutas

* **Estructura de URL:**
  + Explicación de cómo se construyen las URLs en el proyecto
* **Listado de Rutas:**
  + Tabla con las rutas, métodos HTTP, descripción y parámetros

| Ruta | Método HTTP | Descripción | Parámetros |
| --- | --- | --- | --- |
| /addPelicula | GET | Reenvia al formulario de agregar una pelicula |  |
| /updatePelicula | GET | Reenvia al formulario de editar una pelicula |  |
| /checkPelicula | GET | Muestra el detalle de una pelicula especifica |  |
| /deletePelicula | GET | Reenvia a una pagina para confirmar que quiere eliminar una pelicula |  |
| /addPelicula | POST | Recibe los datos del formulario y crea una nueva pelicula |  |
| /updatePelicula | POST | Recibe los datos del formulario y edita una pelicula |  |
| /deletePelicula | POST | Elimina una pelicula de la base de datos |  |
| /addFuncion | GET | Reenvia al formulario de agregar una funcion |  |
| /updateFuncion | GET | Reenvia al formulario de editar una funcion |  |
| /checkFuncion | GET | Muestra el detalle de una funcion especifica |  |
| /deleteFuncion | GET | Reenvia a una pagina para confirmar que quiere eliminar una funcion |  |
| /addFuncion | POST | Recibe los datos del formulario y crea una nueva funcion |  |
| /updateFuncion | POST | Recibe los datos del formulario y edita una funcion |  |
| /deleteFuncion | POST | Elimina una funcion de la base de datos |  |
| /usuarios/{id} | GET | Obtener usuario por ID | id (int) |
| /usuarios | POST | Crear nuevo usuario | nombre, apellido, email |

### 4. Roles de Usuario

* **Tipos de Usuarios:**
  + Hay dos tipos de usuarios, el usuario Admin y el usuario Cliente, el usuario Admin tiene más permisos que el usuario Cliente.
* **Permisos:**
  + Tabla con los permisos asociados a cada rol

| Rol | Permisos |
| --- | --- |
| Administrador | Crear Editar y borrar Películas y Funciones, comprar entradas |
| Usuario | Comprar entradas |

### 5. División de Tareas

* **Asignación de Tareas:**
  + Tabla con las tareas asignadas a cada miembro del equipo

| Tarea | Miembro del Equipo | Estado |
| --- | --- | --- |
| Diseño de la base de datos | Santiago Lopez y Nicolas Speratti | Completada |
| Desarrollo del controlador y vista (servlets, jsp) | Santiago Lopez y Nicolas Speratti | Completada |
| Desarrollo del modelo (clases beans DAO) | Santiago Lopez y Nicolas Speratti | Completada |

### 6. Capas Backend

* **Capa de Presentación:**
  + Utilizamos archivos jsp que mandan peticiones a los controladores
* **Capa de Negocio:**
  + Utilizamos servlets como controladores que interactúan como intermediarios entre las vistas y el modelo
* **Capa de Acceso a Datos:**

Utilizamos Clases DAO, clases beans, y JDBC como conector a base de datos