

UCM CDs

SRS

SUBRAYADO EN ROJO: SE PRETENDE ELIMINAR

SUBRAYADO EN AMARILLO: TERMINADO

SUBRAYADO EN VERDE: se ha modificado de cines a cds

Pablo Camacho Carrasco
Iván Cepeda Álvarez
Antonio Cuenca Bravo
Daniel Fernández Ortiz
Diego Flores Simón
Cristian García Moruno
Ignacio Garro San Millán
Iulian-Nicolae Muntean

TABLA DE CONTENIDO

1. Introducción

- 1.1 Propósito
- 1.2 Alcance
- 1.3 Definiciones, acrónimos y abreviaturas
- 1.4 Referencias
- 1.5 Resumen

2. Descripción general

- 2.1 Perspectiva del producto
- 2.2 Funciones del producto
- 2.3 Características del usuario
- 2.4 Restricciones
- 2.5 Supuestos y dependencias

2.6 Requisitos futuros

3. Requisitos específicos

3.1 Interfaces externos

3.2 Funciones

3.3 Requisitos de rendimiento

3.4 Modelo del dominio (Logical DB Requirements)

3.5 Restricciones de diseño

3.6 Atributos del sistema software

https://drive.google.com/file/d/1_HesF6jrgc94GqsQiBimDkIHTpT32cSp/view?usp=share_link

tabla de control de cambios? después de hacer cambios al documento se registra y se sube a github

Versión	Fecha	Autores	Descripción
1	28/01/2023	Daniel Fernández Ortiz Cristian García Moruno Ignacio Garro Iulian-Nicolae Muntean	Corrección de la mayoría de las notas proporcionadas por el profesor
2	30/01/2023	Daniel Fernández Ortiz Ignacio Garro	Cambio de proyecto de UCM Cines a UCM CDs

1. Introducción

1.1 Propósito

El documento de especificación de requisitos de software (SRS) tiene como propósito establecer la base para un acuerdo entre clientes y desarrolladores sobre lo que debe hacer el software. A través de este documento se consigue establecer la base para un acuerdo entre clientes y desarrolladores sobre qué debe hacer el software, consiguiendo así reducir el esfuerzo de desarrollo, proporcionando una base para la planificación y la verificación.

La SRS tiene como objetivo especificar el “qué” de lo que va a hacer el sistema, no el “cómo”. Es una forma de justificar las acciones para llevar a cabo el desarrollo del proyecto.

1.2 Alcance

Identificaremos al proyecto de software por el nombre de “UCM CDs”.

El objetivo principal de nuestro proyecto es la creación de una aplicación que gestione el negocio “UCM CDs”. Esta aplicación se encargará de gestionar las actividades económicas que realiza esta empresa, como es la gestión de clientes, gestión de ventas y la gestión de productos disponibles. También habrá que facilitar al usuario los productos disponibles que vaya a querer comprar en la aplicación.

A través de esta práctica, podremos proporcionar a “UCM CDs” una aplicación de software de calidad que permitirá controlar y manejar sus actividades económicas de una forma sencilla y eficaz, mejorando la productividad del personal y las ventas del negocio.

1.3 Definiciones, acrónimos y abreviaturas

Software: Instrucciones que forman programas informáticos a través de lenguajes de programación, que serán las instrucciones que dirán a los ordenadores las tareas que deben realizar.

Hardware: Los componentes físicos que constituyen un ordenador o sistema informático.

Interfaces: Es la conexión funcional entre dos sistemas informáticos.

GUI: Es la interfaz gráfica que un usuario experimenta al utilizar una aplicación digital, normalmente representada por texto, imágenes, video y diseño gráfico, con la finalidad de crear una conexión entre la información del sistema informático y el usuario.

Base de Datos: Es el conjunto de datos e información que está organizado y estructurado de tal modo específico, para que su contenido pueda ser tratado y analizado de manera eficiente, sencilla y económica con un programa informático como es SQL.

SQL: Significa Lenguaje de Consulta Estructurada, y es un lenguaje de programación que se especializa en realizar operaciones sobre una base de datos.

JAVA: Es un lenguaje de programación de propósito general que utiliza la programación orientada a objetos.

JDBC: (Java Data Base Connectivity) es una API para llevar a cabo operaciones sobre bases de datos.

API: *Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones

Entidades:

- Películas: productos que se proyectan en el cine.
- Salas: lugares donde se proyectan las películas.
- Clientes: personas que visualizan las películas.
- Ventas: registro de la ventas de entradas.
- Administradores: gestionan las películas y las salas.
- Empresas cinematográficas: organizaciones que proporcionan las películas.

1.4 Referencias

Para llevar a cabo esta SRS, aparte de la teoría que se nos ha aportado en el campus virtual y de haber seguido el estándar IEEE Std. 830-1998, hemos acudido a las siguientes páginas para consultar la información adicional que necesitábamos.

<https://definicion.de/base-de-datos/>

<https://economipedia.com/definiciones/politicas-regulatorias.html>

<https://www.redeszone.net/tutoriales/internet/que-es-protocolo-tls-handshake/>

<https://blogcandidatos.springspain.com/desarrollo-profesional/supuestos-del-proyecto-que-son-y-como-gestionarlos/#:~:text=Las%20suposiciones%20son%20creencias%20basadas,no%20llegar%20a%20buen%20t%C3%A9rmino.>
<https://www.xataka.com/basics/api-que-sirve>

1.5 Resumen

En esta SRS describiremos tanto la descripción general del producto, como el conjunto de requisitos que serán necesarios en la implementación de la aplicación “UCM CDs” para poder cumplir con las necesidades de nuestro cliente. Así podremos crear una aplicación con todas las funcionalidades necesarias para los usuarios de la aplicación, proporcionando una experiencia completa.

Este SRS está organizado usando la estructura proporcionada por nuestros profesores de la UCM.

2. Descripción general

Esta aplicación se va a llevar a cabo, para poder gestionar la tienda de CD 's de la UCM, con el objetivo de agilizar tanto la gestión de información de la empresa para los clientes, como la facilitación de la compra de entradas para los usuarios.

Con ella los administradores serán capaces de acceder a toda la información de su negocio, para poder ser más productivos y tomar mejores decisiones gracias a la nueva herramienta proporcionada. Dispondrán de una lista ordenada de todas las salas con sus respectivos horarios y además la información de contacto de los clientes.

Por otra parte, habrá una interfaz para los usuarios de la aplicación que les aportará toda la información necesaria para poder realizar la compra de productos desde la aplicación web.

2.1 Perspectiva del producto

En cuanto al contexto de la implantación de la aplicación, se llevará a cabo el negocio ficticio “UCM CD’s” para poder facilitar la gestión de la empresa, tanto por el lado del personal del cine, como por el lado de los clientes que facilitará la obtención de información y compra de productos.

- **Interfaces del sistema:**

- El sistema tendrá una interfaz para los clientes y otra diferente para los administradores.

- **Interfaces de usuario:**

- La interfaz está diseñada para poder realizar las necesidades de nuestros usuarios, conectándose a nuestra base de datos de los usuarios y pudiendo realizar las funciones que deseen, siempre y cuando tengan el permiso para hacerlo.

- **Interfaces de administrador:**

- Esta interfaz está diseñada para disponer de toda la información “privada” del cine, además de las mismas funcionalidades que tienen los usuarios.

- **Interfaces hardware:**

- La interfaz hardware está dividida en el ordenador de escritorio que tendrá el negocio para poder administrar sus datos, y en otro ordenador desde donde podrán acceder los clientes.

- **Interfaces software:**

- Para llevar a cabo esta aplicación haremos uso de JAVA y de MYSQL.

- **Interfaces de comunicación:**

- Usaremos JDBC, un API que nos permitirá conectarnos a la base de datos usando JAVA.
- **Memoria:**
 - Todavía no se han tomado en consideración las características o límites de la memoria, debido a nuestra falta de experiencia en tener en cuenta estos parámetros a la hora de crear una aplicación.
- **Operaciones:**
 - Hay dos modos de operación para los usuarios, por un lado está el de los clientes y por otro el del personal. Éste tendrá acceso a más información que le ayudará a gestionar el negocio.
- **Requisitos de adaptación:**
 - En el local se instalarán dos ordenadores con nuestra aplicación, uno para los clientes y otro para el personal. En un futuro se podría adaptar a más dispositivos.

2.2 Funciones del producto

DEBE ESTAR ACORDE AL PUNTO 3.4

Nuestra aplicación tiene como objetivos principales:

- Facilitar al usuario la cartelera de películas.
- Facilitar al usuario la disponibilidad en tiempo real de las salas de cine.
- Facilitar al usuario la venta de entradas a los clientes con los puntos anteriores.

Además, la aplicación se divide en las siguientes entidades/ los siguientes módulos:

- **Productos**
 - Añadir productos: añade un producto.
 - Eliminar productos: deshabilita un producto de la lista.
 - Actualizar productos: actualiza la información de un producto.
 - Buscar producto: muestra el informe de un producto.
 - Listar productos: muestra el informe de todos los productos.

- **Clientes**
 - Alta cliente: añade un cliente nuevo a la base de datos.
 - Baja cliente: deshabilita a un cliente.
 - Actualizar clientes: actualiza el informe del cliente.
 - Buscar clientes: muestra el informe de un cliente.
 - Listar clientes: muestra el informe de todos los clientes.
- **Alquiler**
 - Crear alquiler: asocia un alquiler a un cliente.
 - Cerrar alquiler: guarda los datos y crea la factura.
 - Anular alquiler: calcula el importe de los productos alquilados.
 - Buscar alquiler: muestra el informe de un alquiler.
 - Listar alquileres: muestra un informe de todos los alquileres.
- **Ventas**
 - Crear venta: asocia una venta a un cliente.
 - Cerrar venta: guarda los datos y crea la factura.
 - Devolución venta: calcula el importe de los productos devueltos.
 - Buscar venta: muestra el informe de una venta.
 - Listar ventas: muestra un informe de todas las ventas.
- **Marcas**
 - Añadir marcas: añade una nueva marca.
 - Eliminar marcas: deshabilita una marca.
 - Actualizar marca: actualiza los datos de una marca.
 - Buscar marcas: muestra el informe de una marca.
 - Listar marcas: muestra el informe de todas las marcas.
- **Empleado**
 - Añadir empleado: añade un nuevo empleado.
 - Eliminar empleado: deshabilita a un empleado.
 - Actualizar empleado: actualiza un empleado.
 - Buscar empleado: muestra el informe de un empleado.
 - Listar empleados: muestra el informe de todos los empleados.

2.3 Características del usuario

Habrán dos tipos de usuarios a los que va dirigida la aplicación, por una parte están los usuarios interesados en comprar un producto de la tienda y por otro están los administradores de la aplicación, que será el personal de la tienda que se encargue de gestionar todo lo relacionado con la actividad de la tienda.

Cada usuario tendrá necesidades distintas, y por consiguiente sus requisitos serán distintos. Además los administradores necesitarán gestionar de una forma eficaz y rápida su negocio, así que debemos tener esto en cuenta.

Este personal tendrá experiencia con el negocio y capacidad de adaptarse a las tareas de la aplicación, ya que esta no requiere de un gran nivel educativo o capacidad intelectual. Por otro lado los clientes que quieren comprar una entrada, estarán solo interesados en la información de los productos disponibles y de poder realizar una compra rápida del producto, sin requerir educación del servicio o experiencia anterior. Por lo tanto el proceso de venta de entradas como la gerencia de la empresa, deben ser intuitivos.

2.4 Restricciones

- Políticas de regulación:

- La aplicación se desarrollará utilizando Java IDE, y será una aplicación de escritorio, de software libre, por lo que no será necesario pagar por su uso.

- Limitaciones hardware:

- Para nuestra aplicación no se necesitará de ningún hardware específico para que funcione correctamente, tan solo un PC para poder correr la aplicación y conectarlo una Base de Datos lo suficientemente potente para poder almacenar todos los datos y correr eficientemente.

- Interfaces con otras aplicaciones:

- Nuestra aplicación estará conectada a una base de datos para poder estar actualizado con todos los cambios de información.

- Operaciones en paralelo:

- La aplicación estará pensada para solo operar con un administrador a la vez. Pero en el caso de los usuarios que quieran comprar tickets, se podrá manejar un gran número de usuarios a la vez.

- Funciones de auditoría:

- No se considerarán las auditorías para nuestra aplicación de software.
- Funciones de control:
 - El control de la aplicación se llevará a cabo en un PC, donde se podrá registrar el administrador, o lo podrá acceder un cliente para poder utilizarlo.
- Requisitos de lenguaje de alto nivel:
 - La aplicación se desarrollará utilizando el programa JAVA.
- Protocolos de signal handshake:
 - No se considerarán la seguridad de los datos de nuestros usuarios.
- Requisitos de fiabilidad:
 - La aplicación debe de tener una alta fiabilidad, teniendo un porcentaje bajo de errores, y ser capaz de seguir funcionando cuando aparece un error y no bloquearse.
- Criticidad de la aplicación:
 - Nuestra aplicación no tendrá un uso real en un negocio por lo que no es muy importante. Pero en el caso en el que sí lo fuera, sería crucial para el negocio de la empresa, ya que un software con errores podría causar grandes pérdidas al establecimiento.
- Consideraciones de robustez y seguridad:
 - No se considera la seguridad de nuestra aplicación contra posibles ciber criminales.

2.5 Supuestos y dependencias

El programa sólo estará optimizado para el sistema operativo Windows, así que dispositivos como Mac, quedarán inutilizados.

En caso de que se produzcan múltiples entradas de usuarios o clientes en el sistema, éste será capaz de responder de manera eficaz y no colapsar, pero en el caso de que ocurra cualquier fallo, nuestro equipo será capaz de resolverlo y mantener todo bajo control.

2.6 Requisitos futuros

La meta principal en un futuro es implementar la versión para dispositivos móviles o tablets en la que la interfaz será ligeramente diferente a la Web para Desktop (escritorio). Esto permitirá a los clientes comprar entradas desde cualquier lugar con conexión a internet además de disponer de la cartelera en todo momento.

3. Requisitos específicos

3.1 Interfaces externos

PONER ALGUNOS PANTALLAZOS DE LA APLICACIÓN (habrá que esperarse a terminarla)

- La aplicación contendrá una interfaz que estará formada por elementos de texto, imágenes, márgenes, links, listas, menús y botones, todo con el propósito de poder realizar su función de una forma óptima y correcta.

3.2 Funciones

**DEBE ESTAR ACORDE AL PUNTO 2.2 Y NO DEBERÍAN HABER DETALLES DE LA INTERFAZ DE USUARIO EN LOS DIAGRAMAS DE ACTIVIDAD
DAR COMENTARIOS MÁS ESPECÍFICOS
¡FALTA TERMINAR DE CORREGIR!**

1. Productos

1.1. Añadir Productos

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** añade un producto
- **Entrada:** nombre del producto
- **Salida:** identificador del producto

- **Origen:** interfaz registro y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** añadir un producto a la tienda
- **Precondición:** sin repeticiones de identificadores ni de nombres
- **Postcondición:** sin repeticiones de identificadores ni de nombres
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

1.2. Eliminar Productos

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** media
- **Estabilidad:**
- **Descripción:** deshabilita un producto de la tienda
- **Entrada:** identificador del producto
- **Salida:** mensaje
- **Origen:** interfaz eliminar producto y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** deshabilita un producto de la tienda
- **Precondición:** el nombre introducido debe existir en la base de datos y que esté habilitada en la cartelera
- **Postcondición:** se deshabilita el producto de la tienda
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

1.3. Actualizar Productos

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** media
- **Estabilidad:**
- **Descripción:** actualiza la información de un producto de la tienda
- **Entrada:** identificador del producto
- **Salida:** mensaje
- **Origen:** interfaz actualizar producto y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** deshabilita un producto de la tienda
- **Precondición:** el nombre introducido debe existir en la base de datos y que esté habilitado en la tienda

- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

1.4. Buscar Productos

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:** media
- **Descripción:** busca un producto
- **Entrada:** nombre de el producto
- **Salida:** identificador del producto
- **Origen:** interfaz buscar producto y cliente invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** busca y muestra un producto
- **Precondición:** el nombre del producto debe existir en la base de datos y estar habilitada en la tienda
- **Postcondición:** se muestra información detallada del producto
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

1.5. Listar Productos

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:** media
- **Descripción:** muestra la lista de los producto
- **Entrada:** ninguna
- **Salida:** interfaz lista productos (ninguna)
- **Origen:** interfaz menú y cliente invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** muestra la lista de productos
- **Precondición:** ninguna
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

2. Clientes

2.1. Alta cliente

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** añade un cliente
- **Entrada:** DNI, nombre e identificador de tarjeta
- **Salida:** DNI
- **Origen:** interfaz cliente y cliente invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** añade un cliente
- **Precondición:** sin DNIs repetidos
- **Postcondición:** sin DNIs repetidos
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

2.2. Baja cliente

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** media
- **Estabilidad:**
- **Descripción:** deshabilita un cliente
- **Entrada:** DNI
- **Salida:** datos del cliente
- **Origen:** interfaz clientes
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** deshabilita un cliente
- **Precondición:** el cliente existe
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

2.3. Actualizar cliente

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** media
- **Estabilidad:**
- **Descripción:** actualizar los datos de un cliente determinado
- **Entrada:** DNI
- **Salida:** información del cliente actualizado

- **Origen:** interfaz cliente
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** actualiza los datos del cliente
- **Precondición:** el DNI existe
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

2.4. Buscar cliente

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** busca a un cliente determinado
- **Entrada:** DNI
- **Salida:** muestra los datos del cliente
- **Origen:** interfaz cliente
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** busca a un cliente determinado
- **Precondición:** el cliente existe
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

2.5. Listar clientes

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** muestra un informe de todos los clientes
- **Entrada:** ninguna
- **Salida:** la lista de los cliente
- **Origen:** interfaz clientes
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** muestra un informe de todos los clientes
- **Precondición:** ninguna
- **Postcondición:** ninguna

- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

3. Alquiler

3.1. Crear Alquiler

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** añade un alquiler
- **Entrada:** DNI cliente
- **Salida:** identificador del alquiler
- **Origen:** interfaz alquiler y empleado invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:**
- **Precondición:** no hay identificadores de alquiler repetidos
- **Postcondición:** no hay identificadores de alquiler repetidos
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

3.2. Cerrar Alquiler

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** media
- **Estabilidad:**
- **Descripción:** deshabilitar un alquiler
- **Entrada:** identificador del alquiler
- **Salida:** se muestran los datos de la factura
- **Origen:** interfaz alquiler y empleado invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:**
- **Precondición:** ninguna
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

3.3. Anular Alquiler

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** media

- **Estabilidad:**
- **Descripción:** anular un alquiler
- **Entrada:** identificador del alquiler
- **Salida:** se muestran los datos de la factura
- **Origen:** interfaz alquiler y empleado invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:**
- **Precondición:** existe al menos un alquiler
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

3.4. Buscar Alquiler

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** buscar un alquiler determinado
- **Entrada:** identificador del alquiler
- **Salida:** muestra los datos de la factura
- **Origen:**
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:**
- **Precondición:** existe al menos un alquiler
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

3.5 Listar Alquiler

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** mostrar un informe de todos los alquileres
- **Entrada:** ninguna
- **Salida:** lista de los alquileres
- **Origen:**
- **Destino:** el sistema software
- **Necesita:** base de datos

- **Acciones:**
- **Precondición:** existe al menos un alquiler
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

4. Ventas

4.1. Crear venta

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** se guardan los datos de la venta
- **Entrada:** DNI cliente
- **Salida:** identificador de la venta
- **Origen:** interfaz ventas y empleado invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** se guardan los datos de la venta y se crea una factura
- **Precondición:** no hay identificadores de venta repetidos
- **Postcondición:** no hay identificadores de venta repetidos
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

4.2. Cerrar venta

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** media
- **Estabilidad:**
- **Descripción:** Se guardan los datos y se crea la factura
- **Entrada:** DNI cliente
- **Salida:** se muestran los datos de la factura
- **Origen:** interfaz ventas y empleado invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** se guardan los datos y se crea la factura
- **Precondición:** ninguna
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

4.3. Devolución de venta

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** medio
- **Estabilidad:**
- **Descripción:** se calcula el importe de la entradas devueltas
- **Entrada:** identificador de la venta.
- **Salida:** se muestra el importe a devolver
- **Origen:** interfaz ventas y empleado invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** se calcula el importe de la entradas devueltas
- **Precondición:** la venta existe
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

4.4. Buscar ventas

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** muestra un informe de una venta buscada
- **Entrada:** identificador de la venta
- **Salida:** muestra la información de la venta
- **Origen:** interfaz ventas y empleado invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** muestra un informe de la venta
- **Precondición:** la venta debe existir
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

4.5 Listar ventas

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** Muestra un informe de todas las ventas
- **Entrada:** ninguna

- **Salida:** se muestran todas las fechas
- **Origen:** interfaz ventas y empleado invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** muestra el listado de la ventas
- **Precondición:** debe haber al menos una venta
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

5. Marca

5.1. Añadir Marca

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** añade una marca a la base de datos
- **Entrada:** nombre de la marca
- **Salida:** identificador de la marca
- **Origen:** interfaz país y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:**
- **Precondición:** no debe haber marcas repetidas
- **Postcondición:** no debe haber marcas repetidas
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

5.2. Eliminar Marca

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** media
- **Estabilidad:**
- **Descripción:** deshabilita una marca
- **Entrada:** identificador de la marca
- **Salida:** se muestra un informe de la marca deshabilitado
- **Origen:** interfaz marca y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:**
- **Precondición:** la marca debe existir en la base de datos

- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

5.3. Actualizar Marca

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** medio
- **Estabilidad:**
- **Descripción:** Actualizar los datos de una marca determinada
- **Entrada:** identificador de la marca
- **Salida:** la marca actualizado
- **Origen:** interfaz marca y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** actualiza los datos de la marca
- **Precondición:** la marca debe existir en la base de datos
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

5.4. Buscar Marca

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** Mostrar una marca determinada
- **Entrada:** identificador de la marca
- **Salida:** se muestra un informe de la marca
- **Origen:** interfaz marca y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:**
- **Precondición:** la marca debe existir en la base de datos
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

5.5. Listar Marca

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta

- **Estabilidad:**
- **Descripción:** Mostrar un informe de todas las marcas
- **Entrada:** ninguna
- **Salida:** se muestra la lista de las marcas
- **Origen:** interfaz marca y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:**
- **Precondición:** debe haber al menos una marca en la base de datos
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

6. Empleado

6.1. Añadir Empleado

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** Añadir un empleado
- **Entrada:** DNI del empleado
- **Salida:** identificador del empleado
- **Origen:** interfaz empleado y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** añade el empleado a la base de datos
- **Precondición:** no debe haber identificadores de empleados repetidos
- **Postcondición:** no debe haber identificadores de empleados repetidos
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

6.2. Eliminar Empleado

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** medio
- **Estabilidad:**
- **Descripción:** deshabilita un empleado
- **Entrada:** identificador del empleado
- **Salida:** muestra el nombre del empleado deshabilitado

- **Origen:** interfaz empleado y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** muestra el nombre del empleado deshabilitada
- **Precondición:** el empleado debe existir en la base de datos y estar habilitada
- **Postcondición:** el empleado queda deshabilitada
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

6.3. Actualizar Empleado

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** medio
- **Estabilidad:**
- **Descripción:** Actualizar los datos de un empleado determinada
- **Entrada:** identificador del empleado
- **Salida:** nombre del empleado actualizada
- **Origen:** interfaz empleado y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** actualiza los datos del empleado
- **Precondición:** el empleado debe existir en la base de datos
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

6.4. Buscar Empleado

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** Mostrar el informe de un empleado determinada
- **Entrada:** identificador del empleado
- **Salida:** se muestra un informe del empleado
- **Origen:** interfaz empleado y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** muestra un informe del empleado
- **Precondición:** el empleado debe existir
- **Postcondición:** ninguna

- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

6.5. Listar Empleado

- **Descrita por:** Daniel e Ignacio
- **Prioridad:** alta
- **Estabilidad:**
- **Descripción:** Muestra un informe de todos los empleados
- **Entrada:** ninguna
- **Salida:** se muestra un listado de los empleado
- **Origen:** interfaz empleado y administrador invocador
- **Destino:** el sistema software
- **Necesita:** base de datos
- **Acciones:** muestra un informe de todas los empleado
- **Precondición:** debe existir al menos un empleado en la base de datos
- **Postcondición:** ninguna
- **Efectos laterales:** si no se puede realizar la operación se muestra un mensaje de error explicando el motivo

3.3 Requisitos de rendimiento

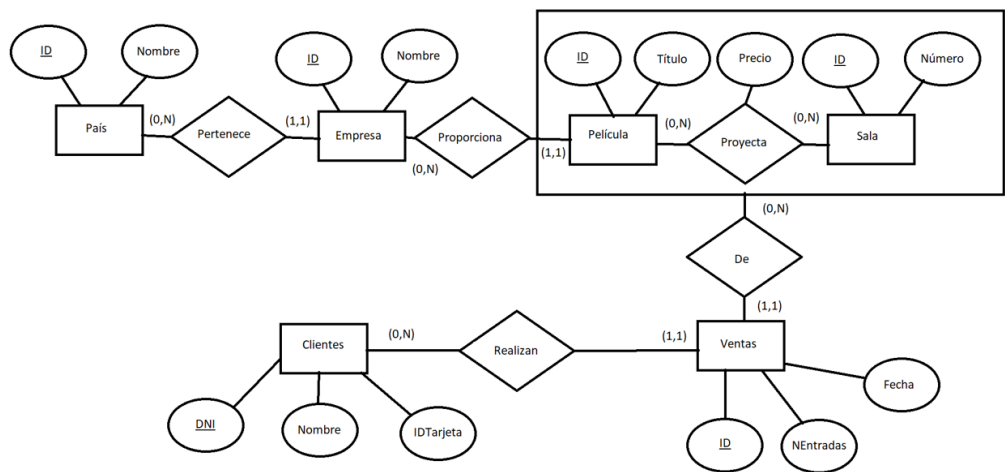
El rendimiento de la aplicación debe de permitirse ser capaz de llevar a cabo sus funciones sin tardar más de 1 segundo por transacción y podrá soportar a un usuario a la vez. El usuario se podrá conectar solamente desde una terminal y cuando pasen más de 10 minutos de inactividad, se desconectará de la aplicación.

3.4 Modelo del dominio (Logical Database Requirements)

SE SUPONE QUE DEBEMOS SACARLO DEL PROYECTO

añadir a salas - aforo

añadir a proyecta - hora



ESTO DE ABAJO CREO QUE SOBRA

ENTIDAD ES	Películas	Salas	Clientes	Ventas	Taquillas	Empresas
Películas	X					
Salas		X				
Clientes			X			
Ventas				X		
Taquillas					X	
Empresas						X

	Menú registro	Registro administra dores	registro usuario	menú administrador	menú usuario	película s	Gestión de cartelera	Administra dor	Compra
Menú registro	X							Ninguno/ varios	
Registro administrador es		X						Ninguno/ varios	
registro usuario	Uno		X		Uno				
menú administrador		Uno		X		Ninguno /Varios	Una	Ninguno/ varios	
menú usuario					X	ninguna /varias			Uno

películas					Ninguno /Varios	X			
Gestión de cartelera						Ninguno /Varios	X	Ninguno/ Varios	
Administrador		Uno		Uno		Ninguno /Varios	Uno	X	Ninguna
Compra					Uno	Ninguna /varias			X

- **Menú registro:** interfaz que se encarga de diferenciar los dos tipos de registros.
- **Registro administradores:** interfaz con dos opciones, una para nuevo administrador y otra para uno ya existente.
- **Registro usuario:** interfaz con dos opciones, una para nuevo usuario y otra para uno ya existente.
- **Menú administrador:** interfaz que muestra las diferentes opciones de administrador.
- **Menú usuario:** interfaz que muestra las diferentes opciones de un usuario.
- **Películas:** se encarga de mostrar las películas, sus horarios, salas y asientos ocupados.
- **Gestión de cartelera:** se encarga de gestionar, modificar o actualizar películas de la cartelera.
- **Administrador:** se encarga de mostrar la información diaria de cada sala.
- **Compra:** se encarga de la gestión de ventas de entradas a los clientes.

3.5 Restricciones de diseño

Restricciones de formato:

El proyecto se ajusta al estándar IEEE Std. 830-1998.

Restricciones de tiempo:

El proyecto tendrá dos entregas:

- 1ª entrega: será el día 16/12/2022 y consistirá en la aportación de los siguientes documentos: Plan del proyecto, Especificación de Requisitos

(SRS), el archivo sobre la planificación temporal con el formato de MS Project, los archivos correspondientes a los diagramas de casos de uso y de actividades UML con el formato IBM Rational Software Architect 9.6, además de un informe del trabajo desarrollado por cada integrante del equipo en el que se califica al resto de integrantes de manera individual.

- 2ª entrega: será el día 5/5/2023 y consistirá en la entrega del producto final.

Restricciones de software:

La aplicación se desarrollará a través del lenguaje de programación Java.

3.6 Atributos del sistema software

Fiabilidad:

- La aplicación ha de tener una tasa de fallos muy baja cuando interactúa con sus usuarios y Base de Datos.

Disponibilidad:

- La aplicación debe de siempre estar disponible para su uso, aunque al depender de servidores para la Base de Datos, se apunta por una disponibilidad del 99%.

Seguridad:

- No se considerará la seguridad de nuestra aplicación contra ciberataques.

Mantenibilidad:

- Gracias a que la aplicación está programada en JAVA, será fácil de expandir y añadir nuevas funcionalidades a la aplicación y también será fácil de expandirlo a nuevos sistemas operativos.

Portabilidad:

- La aplicación puede ser multiplataforma al estar desarrollada en JAVA, haciendo el código reutilizable para nuevos sistemas operativos.

