

Programación – 1º DAM

Proyecto – reserva de espacios

Profesor: Jorge Sánchez

- [1] El proyecto consistirá en la creación de una aplicación completa de reserva de espacios en una empresa. Los requisitos funcionales de la aplicación son los siguientes:

Menú inicial

RF 1 Login de Administrador.

La aplicación permite el acceso a un usuario administrador el cual puede acceder a la aplicación indicando una contraseña

RF 2 Login de Departamento.

La aplicación permite el acceso a un representante de departamento. Simplemente se le pedirá la clave del departamento (por ejemplo: VEN)

A partir de cada Login, lo que ve cada tipo de usuario es diferente.

Menú de administrador

RF 3 Listar departamentos.

Muestra la lista de departamentos (producción, ventas,...) del departamento

RF 4 Añadir departamento

Se pide un nombre (que no se tiene que repetir) y un código (por ejemplo: Código: VEN, Nombre: Ventas)

RF 5 Eliminar departamento

Se pide la clave de un departamento y se elimina del sistema. Las reservas realizadas por ese departamento se retiran de la base de datos

RF 6 Listar salas.

Muestra la lista de salas

RF 7 Añadir sala

Se pide un nombre (que no se tiene que repetir) y un código (por ejemplo: Código: SAJ, Nombre: Sala de Juntas). Al eliminar una sala, automáticamente se eliminarán todas sus reservas

RF 8 Eliminar sala

Se pide la clave de una sala y se elimina del sistema. Las reservas realizadas para esa sala se retiran

RF 9 Listar todas las reservas

Se muestra cada sala y las reservas realizadas,

RF 10 Cerrar sesión.

Permite regresar al menú principal.

Menú de departamento

RF 11 **Añadir reserva.**

Se pide la fecha y hora de la reserva, la sala a reservar y el tiempo que se reserva. Las reservas solo se pueden hacer de 9:00 a 14:00, se reservan horas completas (1,2,3,...) y solo se puede reservar si nadie ha reservado la sala en periodo elegido. la reserva se asigna automáticamente al departamento que reserva

RF 12 **Cancelar reserva**

Se indica la sala y la fecha y hora de inicio de la reserva. Si había reserva de ese departamento a esa hora, la reserva se elimina y la sala pasa a estar libre en ese periodo. Solo puede cancelar el departamento que realizó la reserva.

RF 13 **Listar todas las reservas**

Se muestra cada sala y las reservas realizadas en dichas salas.

RF 14 **Cerrar sesión**

Permite regresar al menú principal

- [2] Todas las lecturas tienen que validar las entradas del usuario. Además, la aplicación tiene que ser amigable y leer los datos de forma cómoda para el usuario.
- [3] Hay que realizar un diagrama de clases completo de la aplicación reflejando propiedades, métodos y relaciones entre clases e interfaces, teniendo en cuenta que los datos de la aplicación se deben de guardar en una base de datos
- [4] **No se pueden utilizar listas (ArrayList, TreeSet,etc.)**
- [5] El código del proyecto (incluido el diagrama de clases) se tiene que subir a un repositorio de GitHub.
- [6] Cada día de trabajo en el aula se requerirá un **commit** en el repositorio de GitHub (de otro modo se penalizará la nota)
- [7] Se debe de realizar la documentación en formato JavaDoc de todas las clases y métodos del proyecto. Esta documentación se debe de guardar en una carpeta llamada **docs**, que cuelgue de la raíz del proyecto y se debe publicar en GitHub Pages
- [8] Se creará un **README.md** con las personas que colaboran en el proyecto, el título del proyecto, la asignatura y la imagen UML con el diagrama de clases
- [9] El proyecto incluirá un archivo llamado **script.sql** que incluirá la creación de las tablas de la base de datos, así como la creación de la base de datos y el usuario y permisos de acceso a dicha base de datos

[10] Baremación

- Diseño y código de la base de datos **[10%]**
- Requisitos funcionales del menú principal y cierres de sesión (RF1, RF2, RF10 y RF14) **[5%]**
- Requisitos funcionales de gestión de departamentos (RF3, RF4 y RF5) **[8%]**
- Requisitos funcionales de gestión de salas (RF6, RF7 y RF8) **[7%]**
- Lista de reservas (RF9 y RF13) **[15%]**
- Requisitos funcionales de gestión de reservas (RF11 y RF12) **[15%]**
- Control de errores **[7%]**
- Diagrama de clases **[8%]**
- Publicación del repositorio en GitHub **[5%]**
- Documentación JavaDoc y publicación en GitHub Pages **[10%]**
- Grado de complejidad **[10%]**

[11] Calificación: Este proyecto puntuará en el apartado de actividades (es la única calificación de este apartado por lo que vale un 30% de la nota de esta evaluación)

[12] Penalizaciones:

- Si no se realiza un **commit** cada día en GitHub:
 - Si falta un día: 0,5 puntos
 - Si faltan dos días: 2 puntos
 - Si faltan más de dos días: 3 puntos
- Por copia o plagio
 - Se valora el proyecto con un cero