INFORME FINAL DE PROYECTO

NOMBRE EQUIPO: Equipo 4

INTEGRANTES EQUIPO DESARROLLADOR:

1. Apellidos Nombres
2. Apellidos Nombres
3. Israel Mollo Vargas
4. Introducción
5. Descripción del problema

*Descripción del problema que resuelve el curso*

*El último párrafo debe incluir descripción del objetivo como equipo*

1. Objetivo de la aplicación

*2.1 Objetivo general*

*2.2 Objetivo equipo (tema(s) asignado(s)*

1. Actores involucrados
2. Objetivos de usuarios
3. Requerimientos funcionales[[1]](#footnote-1)
4. Requerimientos no funcionales[[2]](#footnote-2)

**1. Usabilidad**

* **Interfaz amigable**: El sistema debe ser fácil de usar para todos los actores, con una curva de aprendizaje mínima.

**2. Rendimiento**

* **Tiempo de respuesta**: Las acciones en el sistema deben completarse lo más antes posible en términos de performance para la mayoría de las operaciones.

**3. Seguridad**

* **Autenticación y autorización**: Los usuarios deben autenticarse y tener acceso solo a las funciones relevantes a su rol.
* **Protección de datos**: La información sensible debe ser cifrada y protegida contra accesos no autorizados como la contraseña.

**4. Mantenibilidad**

* **Facilidad de actualización**: El sistema debe ser fácil de actualizar y mantener, con una documentación clara.
* **Modularidad**: Debe estar diseñado en módulos que permitan la modificación o adición de funcionalidades sin afectar el sistema global (fusión de partes).

**5. Disponibilidad**

* **Tiempo de actividad**: El sistema debe estar disponible al menos el 99% del tiempo, con planes de contingencia para fallos.

**6. Pluralidad**

* **Idiomas**: Debe soportar dos idiomas como Ingles y Español para tener un personal variado y que haya una comunicación eficiente.

1. Especificación de casos de uso o historias de usuario ------------------ desde aquí: solo lo concerniente al trabajo de equipo
2. Gestión de notificaciones del personal de limpieza

ANALISIS

1. Lista de clases participantes en cada caso de uso[[3]](#footnote-3)

DISEÑO

1. Diagramas de secuencia por cada caso de uso
2. Diagrama de clases a nivel de diseño[[4]](#footnote-4) *-------------clases de todas sus unidades o módulos*
3. Arquitectura de la aplicación

*(Puntos adicionales: arquitectura del proyecto de curso)*

1. Diagrama de despliegue
2. Diseño de interfaces
3. Debe incluir atributos, métodos y relaciones entre clases. Todos los integrantes del equipo aportan sus clases identificadas.

**Anexo 1**

Backlog

H1: Como Administrador quiero poder registrar nuevos usuarios para contratar nuevo personal y habilitarlos en el sistema.

H2: Como Administrador quiero poder dar de baja al personal para asegurar que empleados que ya no ejercen una labor en el salón de eventos ya no entren al sistema

H3: Como Administrador quiero poder ver la lista del personal para ver a todos los empleados que están registrados.

H4: Como Administrador quiero poder actualizar los datos del personal registrado para modificar los datos personales del empleado.

H5: Como Administrador quiero poder ver los horarios del personal para tener un control de las labores de los empleados.

H6: Como Administrador quiero poder crear tareas para asignarlas al personal de la empresa.

H7: Como Administrador quiero poder modificar las tareas para reasignarlas al personal de la empresa.

H8: Como Administrador quiero poder eliminar tareas para asegurar que no se asigne tareas mal hechas

H9: Como Administrador quiero ver el desempeño del personal en cuanto

1. Lista de casos de uso o historias de usuario. Diagrama de casos de uso. [↑](#footnote-ref-1)
2. Si existen [↑](#footnote-ref-2)
3. Emplear el patrón MVC [↑](#footnote-ref-3)
4. [↑](#footnote-ref-4)