

TAREA #1

Sistema de Gestión de Activos de Software

Introducción

Las aplicaciones de software son un activo esencial de las organizaciones modernas. Con el paso del tiempo, las organizaciones van incrementando considerablemente su ecosistema de aplicaciones y en muchas es necesaria la de repositorios que permitan llevar el control de estas. Un Sistema de Gestión de Activos de Software es un repositorio que lleva el control de éstos.

Requerimientos a desarrollar

Se deberá crear un Sistema Web o de escritorio en .NET 6 (ASP.NET y/o C#) que permita agregar información (que sea almacenada en una Base de Datos SQL Server) y se permita buscar la información de los activos de software. El programa se utilizará a través de interfaz web, consola o interfaz gráfica de escritorio.

El programa deberá facilitarle las funciones al usuario que permitan cumplir con los siguientes requerimientos:

- Cada **aplicación** (software) es descrita por un **código de aplicación**, **número de patente**, **nombre de la aplicación**, **descripción**, **tipo de aplicación** (negocio, utilitaria, otra), **fecha de puesta en producción**, **fecha expiración de licencia**.
- Cada aplicación tiene un **departamento** responsable. Para cada departamento, se debe registrar un **código de departamento**, **nombre de departamento** y **jefe seleccionado de los empleados registrados**. Cada departamento puede ser responsable de más de una aplicación.
- Cada aplicación se ejecuta en un determinado **servidor**. Cada **servidor** tiene un **número de serie**, una **marca**, un **modelo**, **fecha de compra**, **capacidad de procesamiento**, **capacidad de almacenamiento**, **memoria** y un **rol** (production, testing, staging, backup). Un servidor puede servir distintos roles para distintas aplicaciones. Por ejemplo. EL servidor X puede ser el servidor de producción para la aplicación A y el servidor de testing para la aplicación B. Entonces una misma aplicación puede ejecutarse en varios servidores con distintos roles.

- El **Sistema de Gestión** lleva el registro de **errores en producción**. Cada error en producción es originado por una aplicación. De **cada** error se registra: un **identificador de error**, **aplicación**, una **descripción**, **servidor donde ocurrió**, la **fecha**, **hora** y el **impacto** (opciones: **bajo**, **medio**, **alto**, **desconocido**).
- Para cada error que se produzca, se genera un **proyecto de corrección**. Un **proyecto de corrección** tiene **personas involucradas** (gestión empleados) y **puede corregir más de un error a la vez**. De cada **proyecto** se registra: **identificador**, **nombre**, **descripción**, **fecha de inicio**, **fecha de finalización**, **esfuerzo estimado** y **esfuerzo real**. Agrega **personas** con **rol asociado** (**coder**, **supervisor**, **tester**). De cada **persona** se registra la **cédula**, el **nombre**, **apellidos**.

1. Diseño y creación Base de Datos

Considere los anteriores requerimientos y diseñe el **diagrama entidad-relación** que de paso al **diagrama relacional**.

Considerando los requerimientos y el diagrama entidad-relación diseñe la base de datos (diagrama relacional) requerida para implementar dichos requerimientos. Para cada tabla identificada, especifique la **llave primaria**, **llaves candidatas** (en caso que aplique, comentarios en el script), **integridad referencial** y cualquier otro elemento necesario. Para cada **columna**, **indique si** debe ser **única**, **si soporta** o no **nulos** y el **dominio** de cada uno (checks en caso de ser necesario) acorde a los requerimientos indicados. El diseño solicitado debe acercarse al modelo de implementación.

Deben entregar dos modelos de entidad-relación, creados de manera independiente por los integrantes del equipo.

Debe **entregar un script con todas las sentencias SQL necesarias para la recreación de la Base de Datos diseñada** (un único diagrama relacional) **e inserts de registros necesarios para la ejecución de la funcionalidad abajo descrita** (Personas, Tipos de aplicación, Roles de servidor, Nivel de Impacto, Rol en Proyecto).

2. Agregar y Eliminar (aplicativo)

La aplicación debe permitir agregar y eliminar la información (que estará modificando los datos almacenados en la Base de Datos) de los siguientes datos:

- Departamentos (agregar)
- Servidores (agregar)
- Software o aplicaciones (agregar y eliminar)
- Errores de producción (agregar)
- Proyecto de corrección, incluye asociar errores de producción (agregar)
- Persona por proyecto (agregar y eliminar)

3. Detalle de Información (aplicativo)

La aplicación desarrollada deberá permitir el **despliegue de los datos** (campos) de toda la información (tablas) de las siguientes tablas:

- a. **Software o aplicaciones**
- b. **Errores de producción**
- c. **Proyecto de corrección** (deseable con errores y empleados asociados)

4. Salir

Finaliza el programa. Los datos persisten en la Base de Datos.

Aspectos técnicos

1. La tarea es en pareja de trabajo (dos personas).
2. Deberá garantizar que las tablas posean claves que brinden unicidad de los registros.
3. El diseño y creación de la base de datos debe realizar en SQL Server y la aplicación en plataforma .NET.
4. **Sólo se permite un único string de conexión en la aplicación que puede ser modificado para la revisión (utilizar autenticación de SQL Server).**

Documentación

La documentación externa deberá incluir:

- a) Portada.
- b) Manual de usuario: instrucciones de compilación, ejecución y uso.
- c) Pruebas de funcionalidad: incluir screenshots.
- d) Descripción del problema.
- e) Diseño del programa: **dos diagramas ER y un diagrama de la base de datos (Relacional).**
- f) Librerías usadas: conexión a base de datos, etc.
- g) **Análisis de resultados: objetivos** alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).

Evaluación

La tarea tiene un valor de **5%** de la nota final, en el rubro de "Tareas".

Desglose de la evaluación de la tarea:

1. Documentación interna 2ptos.
2. Documentación externa 8 ptos.
3. Diseño y creación de la base de datos (script) 40 ptos.
4. Funcionalidad Aplicativa 40 ptos (Requerimientos a desarrollar consola o interfaz)
5. Revisión de tarea 5 ptos.
6. Hora de Entrega 5 ptos.

Aspectos administrativos

Debe crear un archivo **.zip** ("TC1_Integrante1_Integran2.zip") que contenga únicamente un archivo **info.txt** y 3 carpetas llamadas **documentacion**, **programa y scripts**, en la primera deberá incluir el documento de *word* o pdf solicitado y un archivo txt con el código del programa, en la segunda los archivos y/o carpetas necesarias para la implementación de la parte aplicativa y en la tercera todos los archivos necesarios para la creación de la base de datos (sentencias sql). El archivo **info.txt** debe contener la siguiente información (cualidades):

- a. Nombre del curso
- b. Número de semestre y año lectivo
- c. Nombres de los Estudiantes
- d. Número de carnets de los estudiantes
- e. Número de tarea
- f. Fecha de entrega
- g. Estatus de la entrega (debe ser **CONGRUENTE** con la solución entregada):
[Deplorable | Regular | Buena | MuyBuena | Excelente | Superior]

Entrega

Deberá subir el archivo antes mencionado al TEC Digital en el curso de BASE DE DATOS I GR 60, en la asignación llamada "T1" debajo del rubro de "Tareas". En la evaluación de la Tarea el rubro de "Hora de Entrega" valdrá por 5 puntos de la nota total de la tarea, según la siguiente escala:

- a. Si se entrega antes de las 11:55:55 **PM** del viernes 2 de septiembre de 2022, 5 puntos.
- b. Si se entrega antes de las 11:55:55 **AM** del domingo 4 de septiembre de 2022, 2.5 puntos.
- c. Si se entrega antes de las 11:55:55 **PM** del domingo 4 de septiembre de 2022, 0 puntos.

Después de este punto, **NO SE ACEPTARÁN** más trabajos.

Todo el contenido de cada tarea debe ser 100% original y en caso de plagio los estudiantes tendrán nota cero (Artículo 75 del RREA).