Universidad Mariano Gálvez de Guatemala

Sede Suchitepéquez

Cátedra: Diseño Web

Catedrático: Ing. Axel Aguilar Carrillo

Sistema de Activos Fijos

Fase 1

Estudiante: Jorge Alfredo Enríquez Hernández

Carnet: 3090-14-21095

Mazatenango, Suchitepéquez, 25 de agosto de 2018

Introducción

En la actualidad las empresas son muy competitivas, conforme el paso del tiempo se necesita de nuevas estrategias para que la empresa no se quede estancada. Una afirmación sobre este tema es la que dice: “la empresa se debe actualizar, de lo contrario muere”. Por este motivo ha llegado la hora de mencionar a los sistemas de información que junto con las tecnologías de información han sido una estrategia por la que han optado la mayoría de empresas u organizaciones. Los procesos se realizan con más rapidez y se lleva un mejor control, siempre y cuando el sistema empleado funcione de manera correcta y lo sepan utilizar.

En el siguiente documento se encuentra la información acerca de la creación de un sistema de control de activos fijos para una organización. Iniciando con la parte de análisis únicamente; el diagrama entidad-relación del sistema, scripts e información acerca del motivo por el cual vamos a utilizar las herramientas elegidas para este proyecto.

Sistema de Activos Fijos

(Análisis)

Consta de una aplicación que tenga las funciones necesarias para llevar a cabo procesos relacionados con la gestión de activos fijos de una organización.

Para llevar a cabo este proyecto, se solicitan 4 puntos principales: Gestión de usuarios, Productos, Procesos de activos y Reportes. A partir de estos 4 puntos y de los que están implícitamente dentro de ellos, vamos a iniciar con el análisis requerido.

**Funcionalidades**

**Gestión de Usuarios:** esta parte se refiere a la creación y registro de usuarios además del acceso según restricciones por tipo de usuario; esto quiere decir que tendremos diferentes usuarios con diferentes niveles de acceso en el uso de la aplicación. Esto se manejará con la asignación de privilegios para cada usuario registrado.

**Gestión de Productos:** se refiere a los procesos que se van a manejar con productos que necesite la empresa, al hablar de gestión de producto en una organización se refiere a la creación, distribución o comercialización de un producto, en este caso nos inclinamos para la parte de comercialización. No necesariamente significa que el sistema va a ser usado para realizar ventas, sino que se va a utilizar para hacer compras las cuales van a ser el activo fijo que va a ingresar a la organización.

**Gestión de Tipos de Productos:** derivado de la gestión de productos, este en cambio se necesita para llevar el control de los productos que llegan a la empresa, con el fin de separarlos según su tipo y función. Para posteriormente hacer cálculos como por ejemplo la depreciación de mobiliario y equipo por ejemplo.

**Altas y Bajas de Activos:** en este apartado se llevará a cabo el proceso de dar de alta de un producto, por ejemplo, se ha hace con anterioridad la solicitud de un escritorio; este pertenece a la cuenta contable de mobiliario y equipo. Al llegar a la empresa, se tiene preparado en la bodega y luego al darle de alta en el sistema, podemos enviar el escritorio hacia la oficina que lo necesitaba. Por otro lado, al dar de baja, vemos el tiempo de un equipo que ya se encuentre en mal estado y consecuentemente se debe cambiar, por esto, el sistema deberá poder realizar la operación de darle de baja a este equipo, normalmente se hace cuando ya se tiene el sustituto para no retardar ningún proceso de trabajo.

**Reportes:** por último, la parte de reportes es necesaria para tener toda la información con respecto a los activos que se brindaron al usuario, las altas y bajas de activos de la empresa durante un tiempo determinado. Este requisito es importante para tener la constancia de que los procesos se realizaron de manera correcta.

**Características**

**Lenguaje de Programación:** para realizar la aplicación se asignó el lenguaje php, entonces pasaremos a mencionar las características y beneficios que puede traer la utilización de este lenguaje.

**PHP es un lenguaje totalmente libre y abierto.** Tiene una curva de aprendizaje muy baja, su sintaxis es simple y cumple estándares básicos de la programación orientada a objetos. No son necesarios complejos entornos de desarrollo, que incluso necesitan su propio periodo de aprendizaje.



**MySql:** luego de hacer el análisis respectivo, se toma la decisión que para llevar a cabo el proyecto, vamos a utilizar mysql ya que es un sistema de gestión de base de datos muy conocida en el mundo. Se emplea muy a menudo para los entornos web. Pero por qué usar mysql? Hay ciertas ventajas que ésta nos brinda, las cuales vamos a enlistar para justificar su uso dentro del proyecto.

1. Es Multiplataforma: Para S.O. como Windows, Linux y Mac disponemos de nuestro servidor para instalarlo.
2. Es fácil encontrar ayuda: Al ser una base de datos que se utiliza en multitud de aplicaciones web existen multitud de tutoriales, foros, …. en la red en los que podemos encontrar la información que necesitamos
3. Es fácil de aprender: Simplemente con conocer el estándar de SQL podemos manejar la base de datos MySQL si ningún problema.
4. MySQL es una base de datos ampliamente probada por distintos usuarios y empresas con alto éxito.
5. Menos características, Menos mantenimientos: Realmente esto nos da la ventaja para que un programador cualquiera pueda aprender rápidamente como debe mantener la base de datos para sus aplicaciones. Sin necesidad de ser un experto Administrador en Base de Datos (DBA). Bases de datos como Oracle requieren de DBA para la gestión de su información debido a todas las características que tienes para su administración. En cambio MySQL para el funcionamiento habitual de una aplicación incluye unas características mínimas que nos sirven ampliamente para nuestras aplicaciones sin tener que recurrir a un DBA para que administre la base datos.
6. El coste total de inicio es ampliamente inferior al de Oracle o Microsoft SQL Server. Nosotros vamos a poder crear tantos servidores MySQL como necesitemos sin incurrir en ningún coste alto.
7. MySQL es escalable.

**Servidor:** para este punto daremos a conocer el servidor con el cual vamos a trabajar y el por qué elegirlo.

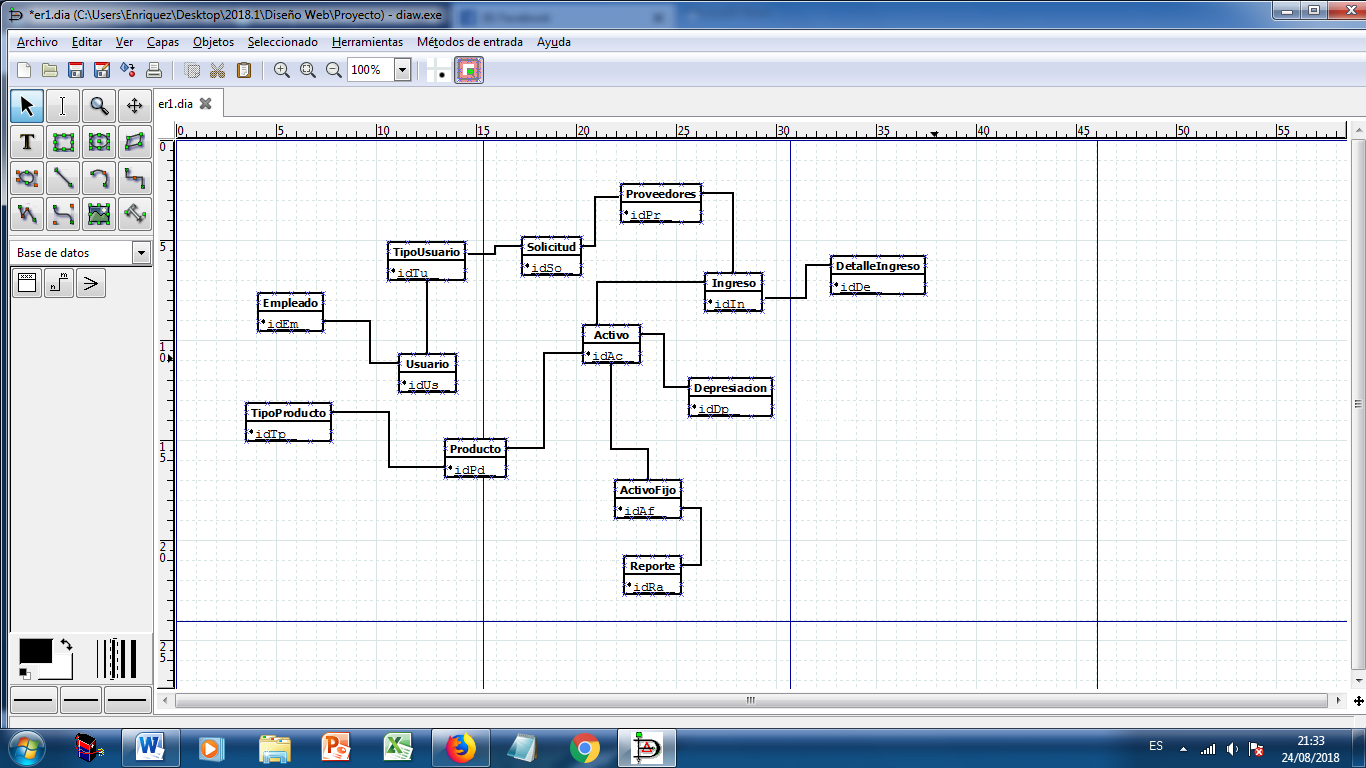
Se opta por XAMMP ya que es un paquete de plataforma libre, es un software que integra en una sola aplicación un servidor web Apache, intérpretes de lenguaje de scripts PHP, un servidor de base de datos MySQL, un servidor de FTP FileZilla, el popular administrador de base de datos escrito en PHP, MySQL, entre otros módulos.



**Estrategia para tener alta disponibilidad:** para este punto, en el cual se necesita que nuestra página o aplicación de encuentre siempre disponible, la opción de hosting es muy buena ya que nos brinda el servicio para poder alojar la aplicación y se pueda acceder a ella en todo momento.



**Modelo ER**



Explicación:

Para el proyecto se necesita un sistema de control de activos fijos, se necesita de hacer gestiones de usuarios y gestión de productos. Por lo que muestra la siguiente explicación del diagrama.

Inicia con la entidad Empleado, la cual está relacionada con Usuario, ya que cada empleado va ser un usuario del sistema. Así mismo, deben de existir diferentes tipos de usuario a los cuales les vamos a asignar distintos privilegios para el uso de la aplicación.

Después encontramos la entidad tipo de usuario conectada con la de solicitud, ya que dependiendo del tipo de usuario, se harán las solicitudes necesarias.

Tenemos 3 entidades relacionadas las cuales son: Solicitud, Proveedores e Ingreso, las cuales inician por el tipo de usuario quien será el encargado de hacer dicha solicitud cuando se necesite producto por ejemplo; este proceso sería el necesario en cuanto a altas de activos. De igual forma Detalle Ingreso va hacer la función de reporte de ingreso.

Luego tenemos la entidad Activo, en donde se tiene la descripción de los activos que tenemos al momento para nuestro proyecto. Sin embargo, se debe hacer gestión de productos, esto lo vamos a tratar principalmente como la adquisición de los productos, por ejemplo si se solicita un escritorio, el cual pasa a formar parte del grupo de activos fijos por eso tenemos la entidad Activo Fijo en donde tendremos la información de los elementos que forman parte de este grupo, con su reporte respectivo.

Producto se necesita por los requerimientos del sistema para hacer gestiones, para eso tenemos la entidad Producto la cual se une con tipo de producto, ambos unidos a la entidad Activo, ya que lo son, y por último están relacionados con la entidad Depreciación para hacer el cálculo de esto en todos los activos que se poseen.

**Scripts**

A continuación se presenta la estructura básica de algunos querys que se necesitan para llevar a cabo el proyecto. Están escritos con la sintaxis SQL debido al uso del gestor de base de datos mencionado en el análisis anterior.

Primero se necesita crear la base de datos que va contener la información para la aplicación, para ello podemos usar la siguiente estructura:

1. CREATE DATABASE Sis\_Activosf;
2. SHOW DATABASES;
3. USE Sis\_Activosf;

Ahora que ya tenemos creada nuestra base de datos, vamos a continuar creando la primera tabla, en este caso será la tabla de Empleado. Como mencionamos, solo agregaremos en este documento ejemplos de la estructura de los objetos, ya que sería redundante explicar cada una de las tablas en las cuales solo cambia el tipo de atributos que tienen.

CREATE TABLE Empleado (

Id NUM (5) not null,

Nombre VARCHAR (20),

Sexo CHAR (1),

Fecha\_de\_nacimiento DATE,

PRIMARY KEY Id);

La estructura anterior sirve para crear cualquier otra tabla. Luego pasamos a insertar datos en una tabla; se realiza de la siguiente forma:

INSERT INTO Empleado

VALUES (‘1’, 'Fede', 'M', '1993-04-12');

Para poder ver la tabla que hemos creado ejecutamos la siguiente instrucción:

SELECT \* FROM Empleado;

A continuación vamos a ver el código de un Trigger muy importante en nuestro proyecto. Trata sobre llevar una auditoria de lo que hace un usuario con las tablas a las que tiene acceso. Primero necesitamos crear la tabla de control, que se va a llamar auditoria, en la cual se van a registrar las acciones que haga el usuario.

CREATE TABLE auditoria(

Usuario varchar2 (25),

Fecha date,

Tabla varchar2 (25),

Operación varchar2 (25);

Trigger Auditoria:

CREATE OR REPLACE TRIGGER tg\_producto

AFTER INSERT OR UPDATE OR DELETE ON Producto

REFERENCING OLD AS old\_buffer NEW AS new\_buffer

FOR EACH ROW

DECLARE

v\_usuario varchar2(25),

v\_fecha date;

BEGIN

SELECT user INTO v\_usuario FROM dual;

SELECT SYSDATE INTO v\_fecha FROM dual;

IF DELETING THEN

INSERT INTO auditoria(usuario,fecha,tabla,operacion)

VALUES (v\_usuario, v\_fecha, 'Producto', 'Elimino');

END IF;

IF INSERTING THEN

INSERT INTO auditoria(usuario,fecha,tabla,operacion)

VALUES (v\_usuario, v\_fecha, 'Producto', 'Inserto');

END IF;

IF UPDATING THEN

INSERT INTO auditoria(usuario,fecha,tabla,operacion)

VALUES (v\_usuario, v\_fecha, 'Producto', 'Actualizo');

END IF;

END;

Este es un tipo de disparador muy importante, ya que nos permite llevar el control de lo que hace el usuario, con esta estructura podemos crear el Trigger para las demás tablas, pero ¿cómo podemos ver eso?

Simplemente mostrando la tabla “auditoria” con la consulta:

SELECT \* FROM auditoria;