MANUAL TÉCNICO PARA EL SISTEMA DE INFORMACIÓN PARA LA EMPRESA SERVITECNICOS DEL CAMPO SAS.

OSCAR ALEJANDRO ACERO SILVA

ANDRES FELIPE TELLEZ CHAVEZ

VICTOR MANUEL OSORIO PARRA

JHOJAN DANIAN ORJUELA BARAHONA

ANGIE XIMENA GONZALEZ BERMUDEZ

FICHA: 2069168

SENA

ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN

FUSAGASUGÁ

2021

MANUAL TÉCNICO PARA EL SISTEMA DE INFORMACIÓN PARA LA EMPRESA SERVITECNICOS DEL CAMPO SAS.

OSCAR ALEJANDRO ACERO SILVA

ANDRES FELIPE TELLEZ CHAVEZ

VICTOR MANUEL OSORIO PARRA

JHOJAN DANIAN ORJUELA BARAHONA

ANGIE XIMENA GONZALEZ BERMUDEZ

INSTRUCTOR:

FERNEY ALEJANDRO RAMIREZ MORA

SENA

ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN

FUSAGASUGÁ

2021

**MANUAL TÉCNICO**

**1. Propósito**

Demostrar toda la estructuración física, técnica, lógica y funcional del sistema de información ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA VENTAS DE MAQUINARIA Y SERVICIOS DE LA INDUSTRIA DEL AGRO PARA LA COMPAÑÍA "SERVITECNICOS DEL CAMPO S.A.S" identificando restricciones, riesgos, respaldos y usuarios que integrarán, proporcionando una guía para las actividades técnicas del sistema en la empresa **SERVITECNICOS DEL CAMPO S.A.**

**2. Introducción**

En este documento se visualizará las especificaciones técnicas del sistema de información creado para la empresa Servitecnicos del Campo S.A.S, la cual facilitará a la persona conocer el requisitos de las herramientas necesarias que debe cumplir para tener acceso al sistema de información, conocer la estructuración y las características que contienen cada uno de los módulos que estarán alojados en el sistema de información para realización de las acciones crear, leer, actualizar y eliminar, y la estructuración de la base de datos.

**2.1. Objetivo general**

Dar a conocer las especificaciones técnicas que debe tener en cuenta para tener un uso del sistema de información con satisfacción.

**2.2. Objetivos específicos**

* Especificar detalladamente los requerimientos que se necesitan para el funcionamiento del sistema de información
* Especificar las características de la base de datos
* Realizar especificaciones de los módulos que contiene el sistema de información
* Realizar una correcta instalación y configuración de las herramientas necesarias

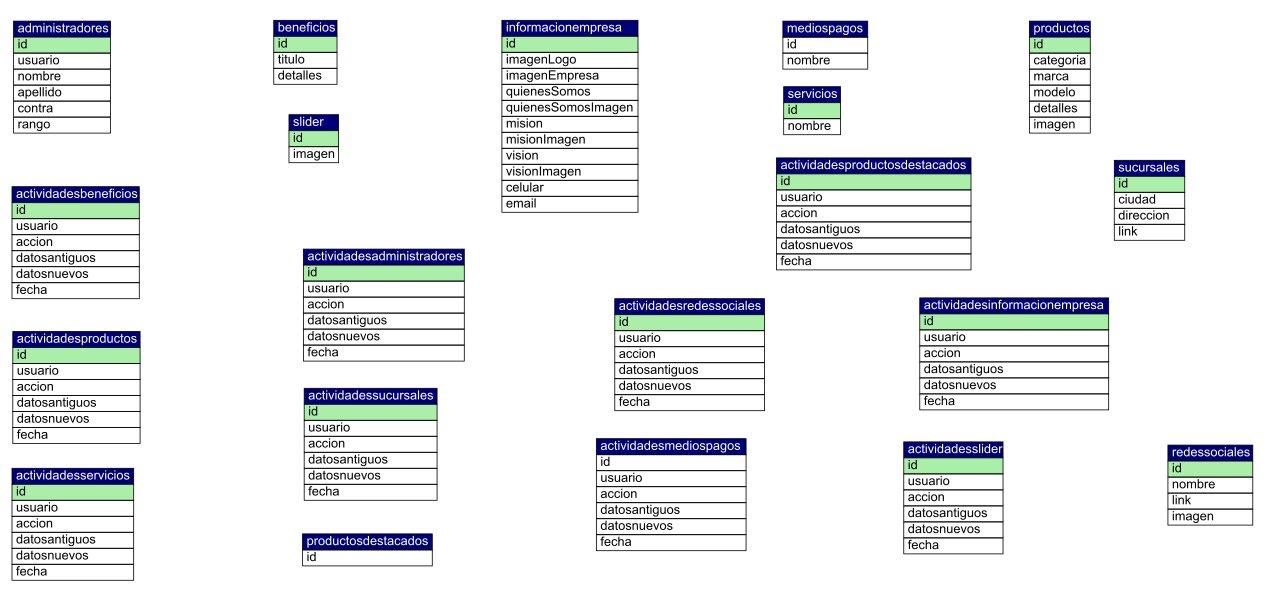
**3. Documentos de referencia**

Los documentos relacionados para la gestión de este manual técnico son:

* Documento de Arquitectura del Sistema.
  + Diagrama de casos de uso **(Punto 6.2 Diagrama uml)**
  + Historial de revisiones “Fecha-version-descripcion-autores.

| **Versión** | **Detalles** | **Fecha** |
| --- | --- | --- |
| v 1.0 | Se codificó la parte visual del sistema de información. | Viernes 9 de abril del 2021 |
| v 1.1 | Agrega las conexiones entre la parte visual y la base de datos del sistema de información. | Viernes 11 de junio del 2021 |
| v 1.2 | Arregla unos bugs de inserción de datos cuando se sobrepasa el límite de paquetes de transferencia hacia la base de datos. | Viernes 6 de agosto del 2021 |
| v 1.3 | Unificación del sistema para un correcto funcionamiento en un host de internet. | Viernes 27 de agosto del 2021 |
| v 1.4 | Estructuración de conexiones variadas por usuario con permisos de acciones definidas. | Viernes 17 de septiembre del 2021 |
| v 1.5 | Arreglo de bugs variados y mejora del código. | Viernes 1 de octubre del 2021 |

* Modelo Relacional - Base de Datos ESD.



**4. Definiciones importantes**

* 1. **Procesos de entrada y salida**

**Entradas**

* Agregar Producto destacado con Botones.
* Agregar Producto.
* Botón para Actualizar y Eliminar Producto.
* Ingresar Información de Empresa.
* Ingresar Imagen Slider.
* Ingresar Información Sucursales.
* Ingresar Información de Servicios.
* Botones para consultar información.
* Botón para Inserción de Información a la Base de Datos.
* Botones para descargar los reportes en Excel.
* Ingresar Información de Redes Sociales.
* Ingresar Información de Servicios.
* Botones para envío de correos masivos.

**Salidas**

* Información plasmada en tablas.
* Información plasmada en Galería de Imágenes.
* Descarga de Reportes en Excel.
* Visualización de Información en el sistema de información como: Imágenes, texto, Productos e Información de la empresa

**5. Descripción de módulos**

**5.1 Módulo Gestión de Cuenta Administrador**

* **Funcionalidad/Propósito:**

Tendrá acceso exclusivamente por el administrador o dueño de la empresa para crear su cuenta de administrador donde se hará su registro por medio de un formulario, para tener acceso a las demás interfaces. De igual manera el administrador puede leer y actualizar sus datos en los respectivos campos expuestos en un formulario.

* **Dependencias funcionales:**

Depende del administrador o dueño para poder realizar su función de creación de usuarios para dar accesoy también depende del índex para iniciar sesión y poder acceder al módulo gestión de cuenta administrador

**5.2 Módulo Gestión de Cuenta Empleado**

* **Funcionalidad/Propósito:**

Tendrá acceso exclusivamente por el administrador o dueño de la empresa para crear las cuentas de los empleados donde se hará su registro por medio de un formulario, para tener acceso a la interfaz de administrador. De igual manera el administrador puede ver los datos de una cuenta de empleado a través de una tabla y de igual manera eliminar un empleado. El empleado podrá ver y actualizar todos los datos de su cuenta de empleado a través de un formulario.

* **Dependencias funcionales:**

Depende del administrador o dueño para poder realizar su función de creación de usuarios para que el empleado tenga accesoy también depende del índex de iniciar sesión para poder acceder al módulo gestión de cuenta administrador y de igual manera a interfaz de administrador.

**5.3 Módulo Gestión de Iniciar Sesión**

* **Funcionalidad/Propósito:**

Tendrá acceso el administrador y empleado que iniciará sesión a través de un formulario para así tener acceso a las interfaces que permitirán la administración del sistema de información.

* **Dependencias funcionales:**

Depende del administrador y empleado para poder realizar su función de iniciar sesión para tener acceso a las interfaces de administrador

**5.4 Módulo Gestión de Productos.**

* **Funcionalidad/Propósito:**

En este módulo los empleados y administradores podrán crear, eliminar y modificar **Productos** mediante un formularioy una Galería de fotos con una ayuda de un buscador con la que visualizamos los productos ya registrados anteriormente con su información correspondiente y presionar el producto deseado para dar continuidad a las acciones de actualizar y eliminar.

* **Dependencias funcionales:** Dependerá de la información suministrada por el administrador o por el empleado para que se pueda visualizar en el sistema de información en el apartado **Maquinarias** y también depende de la interfaz general para poder acceder al módulo gestión de productos.

**5.5 Módulo Gestión de Productos destacados**

* **Funcionalidad/Propósito:**

El módulo dependerá de los usuarios como empleado y/o administrador podrán agregar y eliminar productos a la lista de productos destacados a través de un sistema estilo galería de imágenes junto a la información correspondiente, ver los productos destacados recientemente y los que no están como producto destacado.

* **Dependencias funcionales:**

Dependerá únicamente de la información suministrada por el administrador o por el empleado para que se pueda visualizar en el sistema de información en el apartado **Productos destacados** y también depende de la interfaz de administrador para poder acceder al módulo gestión de Productos destacados.

**5.6 Módulo Gestión de Medios de Pago.**

* **Funcionalidad/Propósito:**

En este módulo los empleados y administradores podrán crear, eliminar y modificar **Medios de pago** mediante un formularioy en una tabla llamada **Medios de Pago Actuales** visualizamos los medios de pago ya registrados anteriormente y poder ver las acciones de actualizar y eliminar.

* **Dependencias funcionales:**

Dependerá únicamente de la información suministrada por el administrador o por el empleado para que se pueda visualizar en el sistema de información en el apartado de **Medios de pago** y también depende de la interfaz general para poder acceder al módulo gestión de Medios de Pago.

**5.7 Módulo Gestión de información de la empresa**

* **Funcionalidad/Propósito:**

Tiene acceso el administrador y el empleado donde realizará las Inserción de datos y modificaciones en la información referente en la empresa mediante un formulario y así visualizar la información registrada anteriormente.

* **Dependencias funcionales:**

Dependerá de la información suministrada por el administrador o por el empleado para que se pueda visualizar en el sistema de información en el apartado **empresa** y también depende de la interfaz general para poder acceder al módulo gestión de información de la empresa.

**5.8 Módulo Gestión de Beneficios.**

* **Funcionalidad/Propósito:**

En este módulo los empleados y administradores podrán crear, eliminar y modificar **Beneficios** mediante un formulario**,** para dar a entender al cliente el **¿Por qué Elegirnos?** y en una tabla llamada **Beneficios Actuales** con la que visualizamos los beneficios ya registrados anteriormente y poder ver las acciones de actualizar y eliminar.

* **Dependencias funcionales:**

Dependerá únicamente de la información suministrada por el administrador o por el empleado para que se pueda visualizar en el sistema de información en el apartado **¿Porque Elegirnos?** y también depende de la interfaz general para poder acceder al módulo gestión de beneficios.

**5.9 Módulo Gestión de Sucursales.**

* **Funcionalidad/Propósito:**

En este módulo los empleados y administradores podrán crear, eliminar y modificar **Sucursales** mediante un formularioy en una tabla llamada **Sucursales Actuales** con la que visualizamos las sucursales ya registradas anteriormente y poder ver las acciones de actualizar y eliminar.

* **Dependencias funcionales:**

Dependerá únicamente de la información suministrada por el administrador o por el empleado para que se pueda visualizar en el sistema de información en el apartado **Empresa/Nuestras Sucursales y** también depende de la interfaz general para poder acceder al módulo gestión de Sucursales.

**5.10 Módulo Gestión de Servicios.**

* **Funcionalidad/Propósito:**

En este módulo los empleados y administradores podrán crear, eliminar y modificar **Servicios** mediante un formularioy en una tabla llamada **Servicios Actuales** con la que visualizamos los servicios ya registrados anteriormente y poder ver las acciones de actualizar y eliminar.

* **Dependencias funcionales:**

Dependerá únicamente de la información suministrada por el administrador o por el empleado para que se pueda visualizar en el sistema de información en el apartado **Servicios y** también depende de la interfaz general para poder acceder al módulo gestión de Servicios.

**5.11 Módulo Gestión de Redes sociales**

* **Funcionalidad/Propósito:**

En este módulo los empleados y administradores podrán crear, eliminar y modificar **Redes Sociales** mediante un formularioy en una tabla llamada **Redes Sociales Actuales** con la que visualizamos las redes sociales ya registradas anteriormente y poder ver las acciones de actualizar y eliminar.

* **Dependencias funcionales:**

Dependerá únicamente de la información suministrada por el administrador o por el empleado para que se pueda visualizar en el footer en el sistema de información **y** también depende de la interfaz general para poder acceder al módulo gestión de Redes sociales.

**5.12 Módulo Gestión del Slider.**

* **Funcionalidad/Propósito:**

Tendrán acceso el administrador y el empleado para realizar las acciones de registrar y actualizar las imágenes que se visualizarán en el slider de la empresa y eliminarlas a través de una galería de fotos visualizada para mayor facilidad de ello.

* **Dependencias funcionales:**

Dependerá de la información suministrada por el administrador o por el empleado para que se pueda visualizar en el índex principal del sistema de información y también depende de la interfaz general para poder acceder al módulo gestión del Slider.

**5.13 Módulo Gestión de Reportes.**

* **Funcionalidad/Propósito:**

Tendrán acceso el administrador y el empleado para realizar las acciones para crear reportes en Excel de algunas secciones como: Beneficios, Medios de pago, Empresa, Productos, Servicios, Sucursales o redes sociales.

* **Dependencias funcionales:**

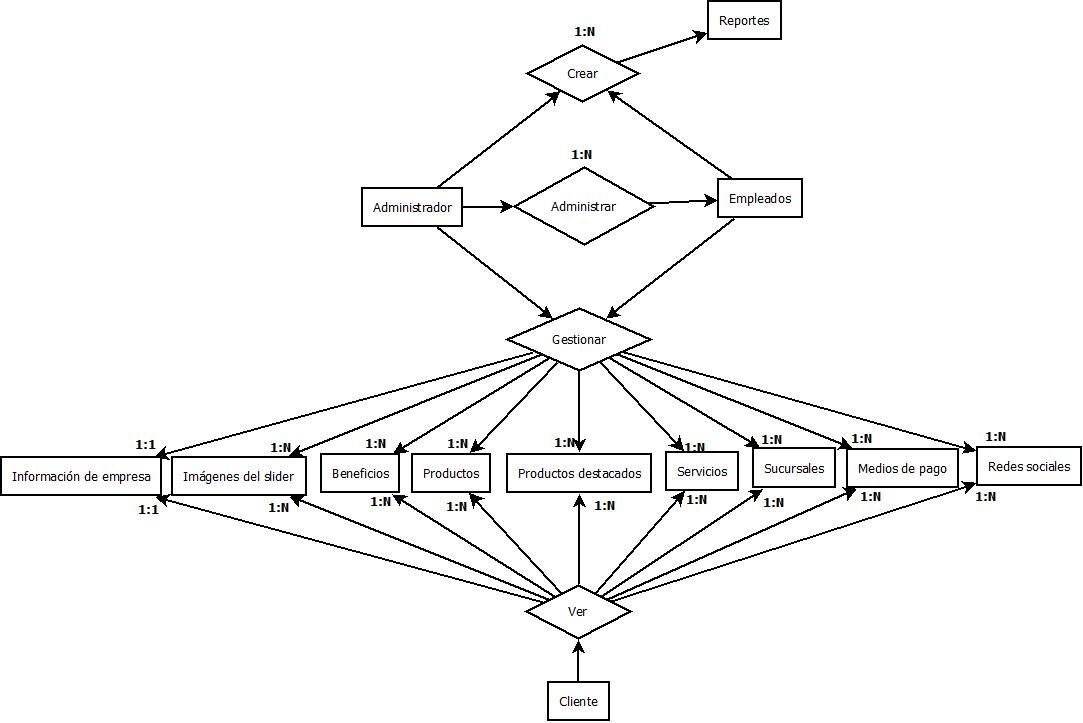
Dependerá de la información suministrada por el administrador o por el empleado en las demás interfaces **obteniendo** información para poder generar los reportes en Excel

**6. Diccionario de datos**

Nombre del documento:

**Base de datos y diccionario de datos.xlsx**

* 1. **Modelo entidad-relación**

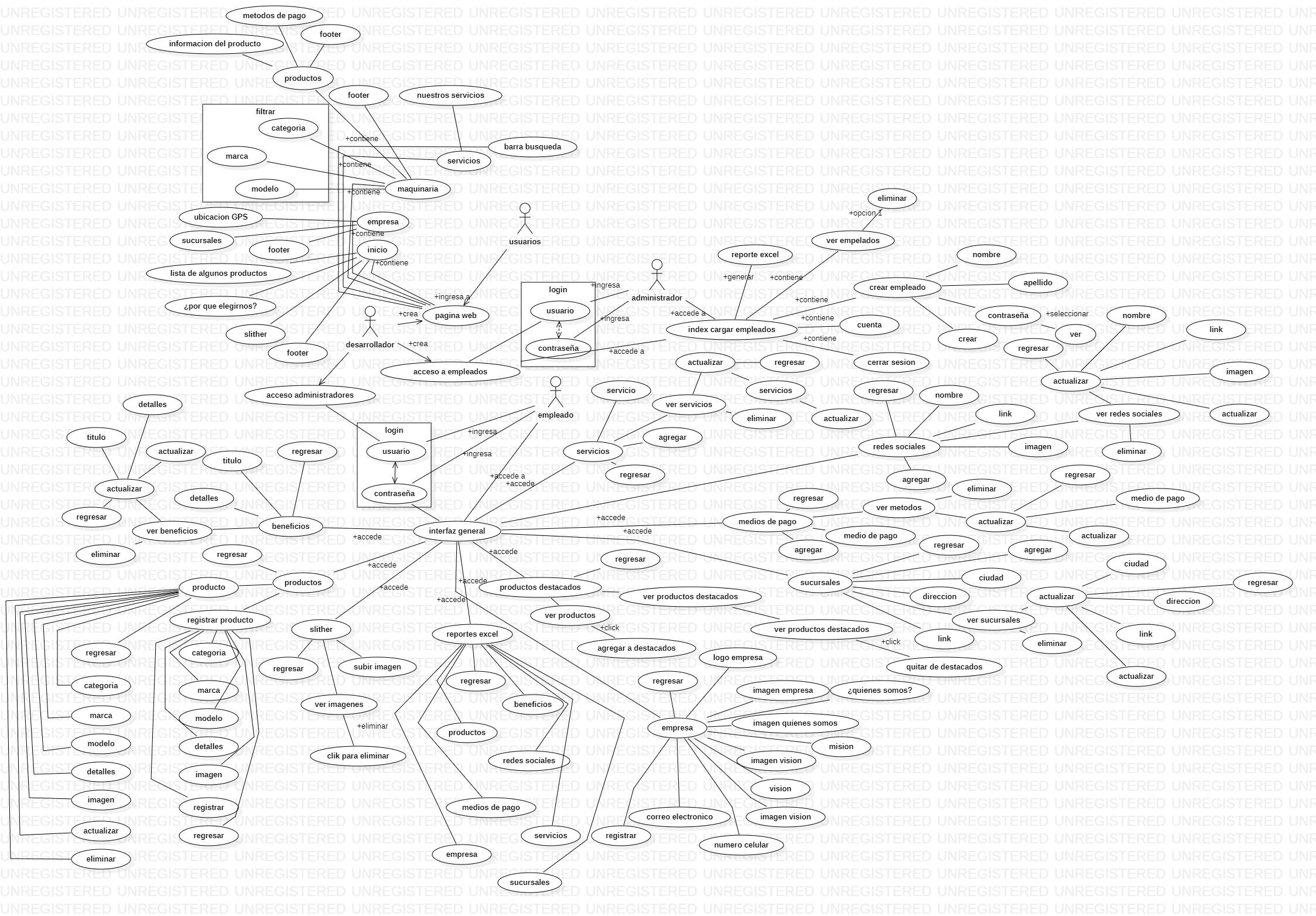
****

* 1. **Diagrama en UML**

Nombre del Documento el cual esta la explicacion de forma individual de cada caso de uso:

**CASOS DE USO.docx**

Diagrama de caso de uso completo:

****

* 1. **Distribución física y lógica de base de datos**

La distribución de la base de datos SERVITECNICOSDELCAMPOSAS a nivel físico y lógico se evidencia en el siguiente listado identificando la arquitectura de la misma:

**Base de Datos servitecnicosdelcamposas**

| **Ubicación del Archivo** | C:\xampp\mysql\data\servitecnicosdelcamposas |
| --- | --- |
| **Codificación Predeterminada** | utf8\_spanish\_ci |
| **Carácter Predeterminado** | utf8 |
| **Número de Tablas** | 20 |
| **Tamaño de la Base de Datos (aprox)** | 30 KB |

**TABLAS DE LA BASE DE DATOS**

**Productos**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\productos.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **6** |
| **Longitud de Fila AVG** | **149254 bytes por dato registrado** |
| **Longitud de Datos** | **597016000 bytes** |
| **Longitud de índice** | **4000** |
| **Longitud máxima de Datos** | **44023414784 bytes** |
| **Datos libres** | **43426398784 bytes** |

**Actividadesproductos**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\actividadesproductos.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **6** |
| **Longitud de Fila AVG** | **1082 bytes por dato registrado** |
| **Longitud de Datos** | **4328000 bytes** |
| **Longitud de índice** | **4000** |
| **Longitud máxima de Datos** | **44023414784 bytes** |
| **Datos libres** | **44019086784 bytes** |

**Productosdestacados**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\productosdestacados.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **1** |
| **Longitud de Fila AVG** | **4 bytes por dato registrado** |
| **Longitud de Datos** | **12 bytes** |
| **Longitud de índice** | **3** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073741812 bytes** |

**Actividadesproductosdestacados**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\actividadesproductosdestacados.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **6** |
| **Longitud de Fila AVG** | **122 bytes por dato registrado** |
| **Longitud de Datos** | **366 bytes** |
| **Longitud de índice** | **3** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073741458 bytes** |

**Sucursales**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\sucursales.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **4** |
| **Longitud de Fila AVG** | **704 bytes por dato registrado** |
| **Longitud de Datos** | **1408 bytes** |
| **Longitud de índice** | **2** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073740416 bytes** |

**Actividadessucursales**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\actividadessucursales.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **6** |
| **Longitud de Fila AVG** | **1.682 bytes por dato registrado** |
| **Longitud de Datos** | **3364 bytes** |
| **Longitud de índice** | **2** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073738460 bytes** |

**Servicios**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\servicios.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **2** |
| **Longitud de Fila AVG** | **54 bytes por dato registrado** |
| **Longitud de Datos** | **270 bytes** |
| **Longitud de índice** | **5** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073741554 bytes** |

**actividadesservicios**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\actividadesservicios.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **6** |
| **Longitud de Fila AVG** | **202 bytes por dato registrado** |
| **Longitud de Datos** | **1010 bytes** |
| **Longitud de índice** | **5** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073740814 bytes** |

**Slider**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\slider.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **2** |
| **Longitud de Fila AVG** | **149011 bytes por dato registrado** |
| **Longitud de Datos** | **447033 bytes** |
| **Longitud de índice** | **3** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073294791 bytes** |

**actividadesslider**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\actividadesslider.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **6** |
| **Longitud de Fila AVG** | **202 bytes por dato registrado** |
| **Longitud de Datos** | **606 bytes** |
| **Longitud de índice** | **3** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073741218 bytes** |

**Redessociales**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\redessociales.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **4** |
| **Longitud de Fila AVG** | **149324 bytes por dato registrado** |
| **Longitud de Datos** | **447972 bytes** |
| **Longitud de índice** | **3** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073293852 bytes** |

**actividadesredessociales**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\actividadesredessociales.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **6** |
| **Longitud de Fila AVG** | **242 bytes por dato registrado** |
| **Longitud de Datos** | **726 bytes** |
| **Longitud de índice** | **3** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073741098 bytes** |

**mediospagos**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\mediospagos.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **2** |
| **Longitud de Fila AVG** | **54 bytes por dato registrado** |
| **Longitud de Datos** | **216 bytes** |
| **Longitud de índice** | **4** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073741608 bytes** |

**actividadesmediospagos**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\actividadesmediospagos.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **6** |
| **Longitud de Fila AVG** | **202 bytes por dato registrado** |
| **Longitud de Datos** | **808 bytes** |
| **Longitud de índice** | **4** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073741016 bytes** |

**informacionempresa**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\informacionempresa.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **11** |
| **Longitud de Fila AVG** | **745974 bytes por dato registrado** |
| **Longitud de Datos** | **745974 bytes** |
| **Longitud de índice** | **1** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1072995850 bytes** |

**actividadesinformacionempresa**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\actividadesinformacionempresa.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **6** |
| **Longitud de Fila AVG** | **2286 bytes por dato registrado** |
| **Longitud de Datos** | **6858 bytes** |
| **Longitud de índice** | **3** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073734966 bytes** |

**beneficios**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\beneficios.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **3** |
| **Longitud de Fila AVG** | **124 bytes por dato registrado** |
| **Longitud de Datos** | **496 bytes** |
| **Longitud de índice** | **4** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073734966 bytes** |

**actividadesbeneficios**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\actividadesbeneficios.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **6** |
| **Longitud de Fila AVG** | **482 bytes por dato registrado** |
| **Longitud de Datos** | **1928 bytes** |
| **Longitud de índice** | **4** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073739896 bytes** |

**administradores**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\administradores.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **6** |
| **Longitud de Fila AVG** | **444 bytes por dato registrado** |
| **Longitud de Datos** | **2664 bytes** |
| **Longitud de índice** | **6** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073739160 bytes** |

**actividadesadministradores**

| **Ubicación del Archivo** | **C:\xampp\mysql\data\servitecnicosdelcamposas\actividadesadministradores.ibd** |
| --- | --- |
| **Motor** | **PHPMYADMIN** |
| **Formato de Fila** | **Compact** |
| **Número de Columnas** | **6** |
| **Longitud de Fila AVG** | **1.066 bytes por dato registrado** |
| **Longitud de Datos** | **6396 bytes** |
| **Longitud de índice** | **6** |
| **Longitud máxima de Datos** | **1073741824 bytes** |
| **Datos libres** | **1073735428 bytes** |

* 1. **Triggers**

Los triggers que se van a mencionar a continuación permitirán gestionar una optimización de algunas funcionalidades permitiendo tener el sistema un poco más robusto. Los disparadores o triggers del sistema de información servitecnicosdelcamposas son los siguientes:

**Trigger De Sucursales.**

**Trigger: Sucursal Agregada.**

* **Nombre del objeto:** Notificar cuando se agregue una nueva sucursal.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se agregue una nueva sucursal al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (Ciudad, dirección y link ) y la fecha actual.
* **Propietario/Esquema:** Tabla sucursales.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `SucursalAgregada` AFTER INSERT ON `sucursales`

FOR EACH ROW INSERT INTO actividadessucursales (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Agregó una sucursal',

'Sin datos',

CONCAT(

'ID:',NEW.id,

' CIUDAD:',NEW.ciudad,

' DIRECCION:',NEW.direccion,

' LINK:',NEW.link

),

NOW())

**Trigger: Sucursal Eliminada.**

* **Nombre del objeto:** Notificar cuando se elimine una sucursal.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se elimine una sucursal del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (Ciudad, dirección y link ) y la fecha actual.
* **Propietario/Esquema:** Tabla sucursales.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `SucursalEliminada` AFTER DELETE ON `sucursales`

FOR EACH ROW INSERT INTO actividadessucursales (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Eliminó una sucursal',

CONCAT(

'ID:',OLD.id,

' CIUDAD:',OLD.ciudad,

' DIRECCION:',OLD.direccion,

' LINK:',OLD.link

),

'Sin datos',

NOW())

**Trigger: Sucursal Modificada.**

* **Nombre del objeto:** Notificar cuando se modifique una sucursal.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se modifique una sucursal al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (Ciudad, dirección y link) y la fecha actual.
* **Propietario/Esquema:** Tabla sucursales.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `SucursalModificada` AFTER UPDATE ON `sucursales`

FOR EACH ROW INSERT INTO actividadessucursales (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Modificó una sucursal',

CONCAT(

'ID:',OLD.id,

' CIUDAD:',OLD.ciudad,

' DIRECCION:',OLD.direccion,

' LINK:',OLD.link

),

CONCAT(

'ID:',NEW.id,

' CIUDAD:',NEW.ciudad,

' DIRECCION:',NEW.direccion,

' LINK:',NEW.link

),

NOW())

**Triggers Sliders**

**Trigger: Slider Agregado.**

* **Nombre del objeto:** Notificar cuando se agregue un Slider.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se agregue un slider al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id) y la fecha actual.
* **Propietario/Esquema:** Tabla sliders.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `SliderAgregado` AFTER INSERT ON `slider`

FOR EACH ROW INSERT INTO actividadesslider (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Agregó una imagen al slider',

'Sin datos',

CONCAT(

'ID:',NEW.id,

' PESO-IMAGEN:',OCTET\_LENGTH(NEW.imagen)

),

NOW())

**Trigger: Slider Eliminado.**

* **Nombre del objeto:** Notificar cuando se elimine un Slider.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se elimine un slider del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id) y la fecha actual.
* **Propietario/Esquema:** Tabla sliders.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `SliderEliminado` AFTER DELETE ON `slider`

FOR EACH ROW INSERT INTO actividadesslider (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Eliminó una imagen del slider',

CONCAT(

'ID:',OLD.id,

' PESO-IMAGEN:',OCTET\_LENGTH(OLD.imagen)

),

'Sin datos',

NOW())

**Trigger: Slider Modificado.**

* **Nombre del objeto:** Notificar cuando se Modifique un Slider.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se modifique un slider del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id) y la fecha actual.
* **Propietario/Esquema:** Tabla sliders.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `SliderModificado` AFTER UPDATE ON `slider`

FOR EACH ROW INSERT INTO actividadesslider (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Modificó una imagen del slider',

CONCAT(

'ID:',OLD.id,

' PESO-IMAGEN:',OCTET\_LENGTH(OLD.imagen)

),

CONCAT(

'ID:',NEW.id,

' PESO-IMAGEN:',OCTET\_LENGTH(NEW.imagen)

),

NOW())

**Trigger Servicios.**

**Trigger: Servicio Agregado.**

* **Nombre del objeto:** Notificar cuando se Agregue un Servicio.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se modifique un servicio del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, nombre) y la fecha actual.
* **Propietario/Esquema:** Tabla servicios.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `ServicioAgregado` AFTER INSERT ON `servicios`

FOR EACH ROW INSERT INTO actividadesservicios (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Agregó un servicio',

'Sin datos',

CONCAT(

'ID:',NEW.id,

' NOMBRE:',NEW.nombre

),

NOW())

**Trigger: Servicio Eliminado.**

* **Nombre del objeto:** Notificar cuando se elimine un Servicio**.**
* **Descripción/Propósito:** El disparador se encargará de generar una notificación cuando se elimine un servicio del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, nombre) y la fecha actual.
* **Propietario/Esquema:** Tabla servicios.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `ServicioEliminado` AFTER DELETE ON `servicios`

FOR EACH ROW INSERT INTO actividadesservicios (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Eliminó un servicio',

CONCAT(

'ID:',OLD.id,

' NOMBRE:',OLD.nombre

),

'Sin datos',

NOW())

**Trigger: Servicio Modificado.**

* **Nombre del objeto:** Notificar cuando se Modifique un Servicio.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se Modifique un servicio del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, nombre) y la fecha actual.
* **Propietario/Esquema:** Tabla servicios.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `ServicioModificado` AFTER UPDATE ON `servicios`

FOR EACH ROW INSERT INTO actividadesservicios (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Modificó un servicio',

CONCAT(

'ID:',OLD.id,

' NOMBRE:',OLD.nombre

),

CONCAT(

'ID:',NEW.id,

' NOMBRE:',NEW.nombre

),

NOW())

**Trigger Red Social.**

**Trigger: Red Social Agregada.**

* **Nombre del objeto:** Notificar cuando se Agregue una Red Social.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se agregue una red social al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, nombre, link) y la fecha actual.
* **Propietario/Esquema:** Tabla redessociales.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `RedSocialAgregada` AFTER INSERT ON `redessociales`

FOR EACH ROW INSERT INTO actividadesredessociales (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Agregó una red social',

'Sin datos',

CONCAT(

'ID:',NEW.id,

' NOMBRE:',NEW.nombre,

' LINK:',NEW.link,

' PESO-IMAGEN:',OCTET\_LENGTH(NEW.imagen)

),

NOW())

**Trigger: Red Social Eliminada.**

* **Nombre del objeto:** Notificar cuando se Elimine una Red Social.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se elimine una red social del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, nombre, link) y la fecha actual..
* **Propietario/Esquema:** Tabla redessociales.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `RedSocialEliminada` AFTER DELETE ON `redessociales`

FOR EACH ROW INSERT INTO actividadesredessociales (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Eliminó una red social',

CONCAT(

'ID:',OLD.id,

' NOMBRE:',OLD.nombre,

' LINK:',OLD.link,

' PESO-IMAGEN:',OCTET\_LENGTH(OLD.imagen)

),

'Sin datos',

NOW())

**Trigger: Red Social Modificada.**

* **Nombre del objeto:** Notificar cuando se Modifique una Red Social.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se modifique una red social del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, nombre, link) y la fecha actual.
* **Propietario/Esquema:** Tabla redessociales.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `RedSocialModificada` AFTER UPDATE ON `redessociales`

FOR EACH ROW INSERT INTO actividadesredessociales (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Modificó una red social',

CONCAT(

'ID:',OLD.id,

' NOMBRE:',OLD.nombre,

' LINK:',OLD.link,

' PESO-IMAGEN:',OCTET\_LENGTH(OLD.imagen)

),

CONCAT(

'ID:',NEW.id,

' NOMBRE:',NEW.nombre,

' LINK:',NEW.link,

' PESO-IMAGEN:',OCTET\_LENGTH(NEW.imagen)

),

NOW())

**Trigger Producto Destacado.**

**Trigger: Producto Destacado agregado.**

* **Nombre del objeto:** Notificar cuando se Agregue un Producto Destacado.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se agregue un producto destacado al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id) y la fecha actual.
* **Propietario/Esquema:** Tabla productosdestacados.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `ProductoDestacadoAgregado` AFTER INSERT ON `productosdestacados`

FOR EACH ROW INSERT INTO actividadesproductosdestacados (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Agregó un producto destacado',

'Sin datos',

CONCAT(

'ID:',NEW.id

),

NOW())

**Trigger: Producto Destacado Eliminado.**

* **Nombre del objeto:** Notificar cuando se Elimine un Producto Destacado.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se elimine un producto destacado del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id) y la fecha actual.
* **Propietario/Esquema:** Tabla productosdestacados.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `ProductoDestacadoEliminado` AFTER DELETE ON `productosdestacados`

FOR EACH ROW INSERT INTO actividadesproductosdestacados (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Eliminó un producto destacado',

CONCAT(

'ID:',OLD.id

),

'Sin datos',

NOW())

**Trigger: Producto Destacado Modificado.**

* **Nombre del objeto:** Notificar cuando se Modifique un Producto Destacado.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se modifique un producto del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id) y la fecha actual.
* **Propietario/Esquema:** Tabla productosdestacados.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `ProductoDestacadoModificado` AFTER UPDATE ON `productosdestacados`

FOR EACH ROW INSERT INTO actividadesproductosdestacados (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Modificó un producto destacado',

CONCAT(

'ID:',OLD.id

),

CONCAT(

'ID:',NEW.id

),

NOW())

**Trigger Producto.**

**Trigger: Producto Agregado.**

* + **Nombre del objeto:** Notificar cuando se Agregue un Producto.
  + **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se agregue un producto del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, categoria, marca, modelos, detalles) y la fecha actual.
  + **Propietario/Esquema:** Tabla productos.
  + **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `ProductoAgregado` AFTER INSERT ON `productos`

FOR EACH ROW INSERT INTO actividadesproductos (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Agregó un producto',

'Sin datos',

CONCAT(

'ID:',NEW.id,

' CATEGORIA:',NEW.categoria,

' MARCA:',NEW.marca,

' MODELO:',NEW.modelo,

' DETALLES:',NEW.detalles,

' PESO-IMAGEN:',OCTET\_LENGTH(NEW.imagen)

),

NOW())

**Trigger: Producto Eliminado.**

* **Nombre del objeto:** Notificar cuando se Elimine un Producto.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se elimine un producto del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, categoria, marca, modelos, detalles) y la fecha actual.
* **Propietario/Esquema:** Tabla productos.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `ProductoEliminado` AFTER DELETE ON `productos`

FOR EACH ROW INSERT INTO actividadesproductos (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Eliminó un producto',

CONCAT(

'ID:',OLD.id,

' CATEGORIA:',OLD.categoria,

' MARCA:',OLD.marca,

' MODELO:',OLD.modelo,

' DETALLES:',OLD.detalles,

' PESO-IMAGEN:',OCTET\_LENGTH(OLD.imagen)

),

'Sin datos',

NOW())

**Trigger: Producto Modificado.**

* **Nombre del objeto:** Notificar cuando se Modifique un Producto.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se modifique un producto del sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, categoria, marca, modelos, detalles) y la fecha actual.
* **Propietario/Esquema:** Tabla productos.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `ProductoModificado` AFTER UPDATE ON `productos`

FOR EACH ROW INSERT INTO actividadesproductos (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Modificó un producto',

CONCAT(

'ID:',OLD.id,

' CATEGORIA:',OLD.categoria,

' MARCA:',OLD.marca,

' MODELO:',OLD.modelo,

' DETALLES:',OLD.detalles,

' PESO-IMAGEN:',OCTET\_LENGTH(OLD.imagen)

),

CONCAT(

'ID:',NEW.id,

' CATEGORIA:',NEW.categoria,

' MARCA:',NEW.marca,

' MODELO:',NEW.modelo,

' DETALLES:',NEW.detalles,

' PESO-IMAGEN:',OCTET\_LENGTH(NEW.imagen)

),

NOW())

**Trigger Medios de Pago.**

**Trigger: Medio de Pago Agregado.**

* **Nombre del objeto:** Notificar cuando se Agregue un Medio de Pago.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se agregue un medio de pago al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id,nombre) y la fecha actual.
* **Propietario/Esquema:** Tabla mediospago.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `MedioPagoAgregado` AFTER INSERT ON `mediospagos`

FOR EACH ROW INSERT INTO actividadesmediospagos (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Agregó un medio de pago',

'Sin datos',

CONCAT(

'ID:',NEW.id,

' NOMBRE:',NEW.nombre

),

NOW())

**Trigger: Medio de Pago Eliminado.**

* **Nombre del objeto:** Notificar cuando se Elimine un Medio de Pago.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se elimine un medio de pago a la sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id,nombre) y la fecha actual.
* **Propietario/Esquema:** Tabla mediospago.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `MedioPagoEliminado` AFTER DELETE ON `mediospagos`

FOR EACH ROW INSERT INTO actividadesmediospagos (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Eliminó un medio de pago',

CONCAT(

'ID:',OLD.id,

' NOMBRE:',OLD.nombre

),

'Sin datos',

NOW())

**Trigger: Medio de Pago Modificado.**

* **Nombre del objeto:** Notificar cuando se Modifique un Medio de Pago**.**
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se modifique un medio de pago al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id,nombre) y la fecha actual.
* **Propietario/Esquema:** Tabla mediospago.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `MedioPagoEliminado` AFTER DELETE ON `mediospagos`

FOR EACH ROW INSERT INTO actividadesmediospagos (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Eliminó un medio de pago',

CONCAT(

'ID:',OLD.id,

' NOMBRE:',OLD.nombre

),

'Sin datos',

NOW())

**Trigger Información Empresa.**

**Trigger: Información Empresa Agregado.**

* **Nombre del objeto:** Notificar cuando se Agregue Información Empresa.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se Agregue información empresa al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id) y la fecha actual.
* **Propietario/Esquema:** Tabla informacionempresa.

**Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `InformaciónAgregada` AFTER INSERT ON `informacionempresa`

FOR EACH ROW INSERT INTO actividadesinformacionempresa (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Agregó la información de la empresa',

'Sin datos',

CONCAT(

'ID:',NEW.id,

' PESO-IMAGEN-LOGO:',OCTET\_LENGTH(NEW.imagenLogo),

' PESO-IMAGEN-EMPRESA:',OCTET\_LENGTH(NEW.imagenEmpresa),

' QUIENES-SOMOS:',NEW.quienesSomos,

' PESO-QUIENES-SOMOS-IMAGEN:',OCTET\_LENGTH(NEW.quienesSomosImagen),

' MISION:',NEW.mision,

' PESO-MISION-IMAGEN:',OCTET\_LENGTH(NEW.misionImagen),

' VISION:',NEW.vision,

' PESO-VISION-IMAGEN:',OCTET\_LENGTH(NEW.visionImagen),

' CELULAR:',NEW.celular,

' EMAIL:',NEW.email

),

NOW())

**Trigger: Información Empresa Eliminada.**

* **Nombre del objeto**: Notificar cuando se Elimine Información Empresa.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se elimine información empresa al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id) y la fecha actual.
* **Propietario/Esquema:** Tabla informacionempresa.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `InformacionEliminada` AFTER DELETE ON `informacionempresa`

FOR EACH ROW INSERT INTO actividadesinformacionempresa (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Eliminó la información de la empresa',

CONCAT(

'ID:',OLD.id,

' PESO-IMAGEN-LOGO:',OCTET\_LENGTH(OLD.imagenLogo),

' PESO-IMAGEN-EMPRESA:',OCTET\_LENGTH(OLD.imagenEmpresa),

' QUIENES-SOMOS:',OLD.quienesSomos,

' PESO-QUIENES-SOMOS-IMAGEN:',OCTET\_LENGTH(OLD.quienesSomosImagen),

' MISION:',OLD.mision,

' PESO-MISION-IMAGEN:',OCTET\_LENGTH(OLD.misionImagen),

' VISION:',OLD.vision,

' PESO-VISION-IMAGEN:',OCTET\_LENGTH(OLD.visionImagen),

' CELULAR:',OLD.celular,

' EMAIL:',OLD.email

),

'Sin datos',

NOW())

**Trigger: Información Empresa Modificada.**

* **Nombre del objeto:** Notificar cuando se Elimine Información Empresa.
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se modifique información empresa al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id) y la fecha actual.
* **Propietario/Esquema:** Tabla informacionempresa.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `InformaciónModificada` AFTER UPDATE ON `informacionempresa`

FOR EACH ROW INSERT INTO actividadesinformacionempresa (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Modificó la información de la empresa',

CONCAT(

'ID:',OLD.id,

' PESO-IMAGEN-LOGO:',OCTET\_LENGTH(OLD.imagenLogo),

' PESO-IMAGEN-EMPRESA:',OCTET\_LENGTH(OLD.imagenEmpresa),

' QUIENES-SOMOS:',OLD.quienesSomos,

' PESO-QUIENES-SOMOS-IMAGEN:',OCTET\_LENGTH(OLD.quienesSomosImagen),

' MISION:',OLD.mision,

' PESO-MISION-IMAGEN:',OCTET\_LENGTH(OLD.misionImagen),

' VISION:',OLD.vision,

' PESO-VISION-IMAGEN:',OCTET\_LENGTH(OLD.visionImagen),

' CELULAR:',OLD.celular,

' EMAIL:',OLD.email

),

CONCAT(

'ID:',NEW.id,

' PESO-IMAGEN-LOGO:',OCTET\_LENGTH(NEW.imagenLogo),

' QUIENES-SOMOS:',NEW.quienesSomos,

' PESO-QUIENES-SOMOS-IMAGEN:',OCTET\_LENGTH(NEW.quienesSomosImagen),

' MISION:',NEW.mision,

' PESO-MISION-IMAGEN:',OCTET\_LENGTH(NEW.misionImagen),

' VISION:',NEW.vision,

' PESO-VISION-IMAGEN:',OCTET\_LENGTH(NEW.visionImagen),

' CELULAR:',NEW.celular,

' EMAIL:',NEW.email

),

NOW())

**Trigger Beneficios.**

**Trigger: Beneficio Agregado.**

* **Nombre del objeto:** Notificar cuando se Agregue un Beneficio.
* **Descripción/Propósito:** El disparador se encargará de generar una notificación cuando se agregue un beneficio al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, titulo, detalles) y la fecha actual.

**Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `BeneficioAgregado` AFTER INSERT ON `beneficios`

FOR EACH ROW INSERT INTO actividadesbeneficios (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Agregó un beneficio',

'Sin datos',

CONCAT(

'ID:',NEW.id,

' TUTULO:',NEW.titulo,

' DETALLES:',NEW.detalles

),

NOW())

**Trigger: Beneficio Eliminado.**

* **Nombre del objeto:** Notificar cuando se Elimine un Beneficio.
* **Descripción/Propósito:** El disparador se encargará de generar una notificación cuando se elimine un beneficio al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, titulo, detalles) y la fecha actual.
* **Propietario/Esquema:** Tabla beneficios.

**Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `BeneficioEliminado` AFTER DELETE ON `beneficios`

FOR EACH ROW INSERT INTO actividadesbeneficios (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Eliminó un beneficio',

CONCAT(

'ID:',OLD.id,

' TUTULO:',OLD.titulo,

' DETALLES:',OLD.detalles

),

'Sin datos',

NOW())

**Trigger: Beneficio Modificado.**

* **Nombre del objeto:** Notificar cuando se Modifique un Beneficio.
* **Descripción/Propósito:** El disparador se encargará de generar una notificación cuando se modifique un beneficio al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, titulo, detalles) y la fecha actual.
* **Propietario/Esquema:** Tabla beneficios.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `BeneficioModificado` AFTER UPDATE ON `beneficios`

FOR EACH ROW INSERT INTO actividadesbeneficios (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Modificó un beneficio',

CONCAT(

'ID:',OLD.id,

' TUTULO:',OLD.titulo,

' DETALLES:',OLD.detalles

),

CONCAT(

'ID:',NEW.id,

' TUTULO:',NEW.titulo,

' DETALLES:',NEW.detalles

),

NOW())

**Trigger Administradores.**

**Trigger: Cuenta Creada.**

* **Nombre del objeto: Notificar cuando se Agregue una Cuenta.**
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se agregue una cuenta al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, usuario, nombre, apellido, contra, rango) y la fecha actual.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `CuentaCreada` AFTER INSERT ON `administradores`

FOR EACH ROW INSERT INTO actividadesadministradores (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Creó una nueva cuenta', 'Sin datos', CONCAT(

'ID:',NEW.id,

' USUARIO:',NEW.usuario,

' NOMBRE:',NEW.nombre,

' APELLIDO:',NEW.apellido,

' CONTRA:',NEW.contra,

' RANGO:',NEW.rango

),NOW())

**Trigger: Cuenta Eliminada.**

* **Nombre del objeto: Notificar cuando se Elimina una Cuenta.**
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se elimine una cuenta al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, usuario, nombre, apellido, contra, rango) y la fecha actual.
* **Propietario/Esquema:** Tabla Administradores.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `CuentaEliminada` AFTER DELETE ON `administradores`

FOR EACH ROW INSERT INTO actividadesadministradores (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Eliminó una cuenta',

CONCAT(

'ID:',OLD.id,

' USUARIO:',OLD.usuario,

' NOMBRE:',OLD.nombre,

' APELLIDO:',OLD.apellido,

' CONTRA:',OLD.contra,

' RANGO:',OLD.rango

),

'Sin datos',

NOW())

**Trigger: Cuenta Modificada.**

* **Nombre del objeto: Notificar cuando se Modifique una Cuenta.**
* **Descripción/Propósito:** El disparador se encargará de crear un registro cuando se modifique una cuenta al sistema de información. Guardará los datos del usuario, acción, datosantiguos y datosnuevos (id, usuario, nombre, apellido, contra, rango) y la fecha actual.
* **Propietario/Esquema:** Tabla Administradores.
* **Código del objeto (cuando no sea confidencial):**

CREATE TRIGGER `CuentaModificada` AFTER UPDATE ON `administradores`

FOR EACH ROW INSERT INTO actividadesadministradores (usuario, accion, datosantiguos, datosnuevos, fecha) VALUES (USER(), 'Modificó una cuenta',

CONCAT(

'ID:',OLD.id,

' USUARIO:',OLD.usuario,

' NOMBRE:',OLD.nombre,

' APELLIDO:',OLD.apellido,

' CONTRA:',OLD.contra,

' RANGO:',OLD.rango

),

CONCAT(

'ID:',NEW.id,

' USUARIO:',NEW.usuario,

' NOMBRE:',NEW.nombre,

' APELLIDO:',NEW.apellido,

' CONTRA:',NEW.contra,

' RANGO:',NEW.rango

),

NOW())

* 1. **Restricciones especiales**
* No se pueden eliminar productos destacados hasta que no se elimine el producto relacionado de la base de datos

**7. Políticas de respaldo**

* 1. **Base de datos**

De acuerdo con el nivel de alcance de la Base de Datos del sistema de Información “ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA VENTAS DE MAQUINARIA Y SERVICIOS DE LA INDUSTRIA DEL AGRO PARA LA COMPAÑÍA "SERVITECNICOS DEL CAMPO S.A.S", se dará un respaldo permanente diario para que se puedan mantener y guardar los datos en caso de alguna pérdida. Este proceso de respaldo lo hace el proveedor del hosting.

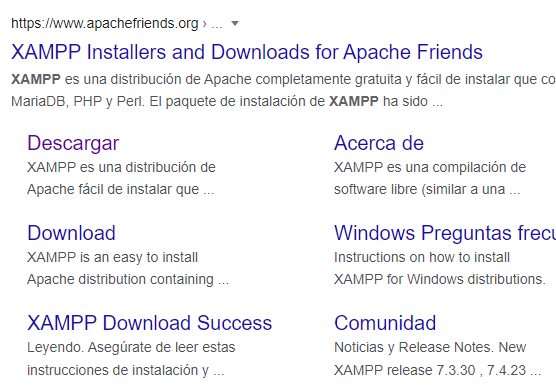
**8. Descripción de interfaces con otros sistemas**

El sistema de información es alojado hacia un servidor, por lo tanto, no tendrá interacción con otros sistemas de información.

**9. Instalación y configuración**

Para este caso se necesitará el programa XAMPP para que se pueda abrir el sistema de información.

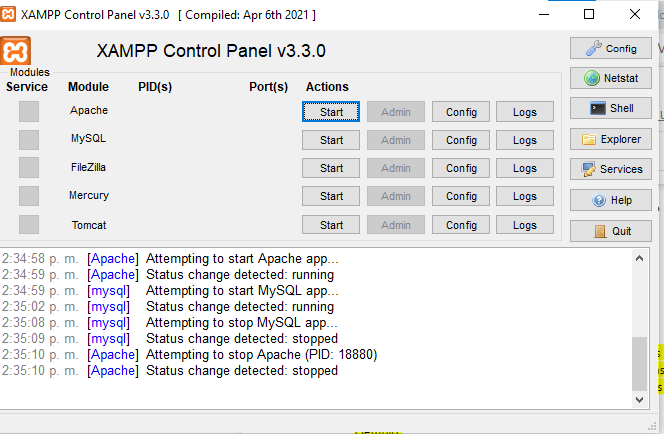
Para este primer proceso de instalación buscaremos el programa para la descarga en Google



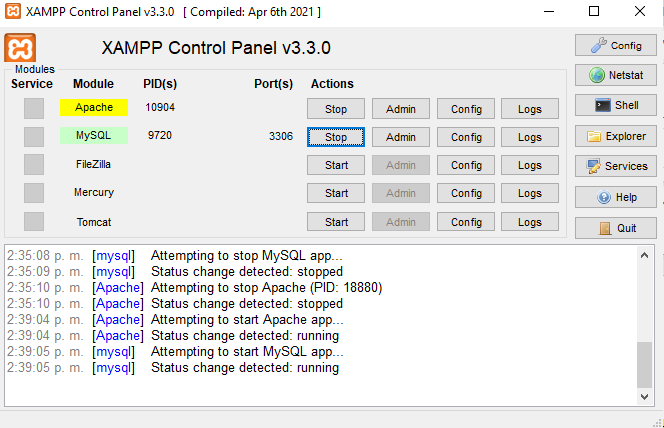
Seleccionamos el apartado descarga.

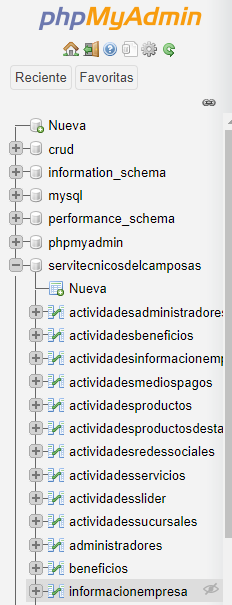


Se escoge la versión que desee instalar dependiendo el Windows y se procede a la instalación del programa para su uso.



Una vez allí se selecciona el apache y el MySQL se les da Start a ambos para que quede activada La opción para visualizar la base de datos o la sistema de información





y ya quedaría el apartado del XAMPP para visualizar la base de datos y el sistema de información.

* 1. **Requisitos generales pre-instalación.**

**Hardware.**

Adaptadores o chips de red, Acceso a internet, Procesador de 2.80GHz o superior, Memoria mínima de RAM 500Mb, Mouse (computador), Teclado (computador) y

Monitor o Display.

**Software.**

Necesario el software que contenga las aplicaciones necesarias para la ejecución como: Navegadores, xampp owamp, wifi, acceso a internet, link o enlace de acceso al sistema de información.

**Sistema Operativo.**

Se necesita un sistema operativo de Windows 7 pero cualquier Windows superior o igual a 7 es funcional para ejecutar lo debido.

**Actualizaciones.**

Se necesitará la actualización del XAMPP a la última versión, también se va a necesitar la última actualización de Chrome o el navegador que utilices.

**Orden de Instalación.**

Primero debemos instalar un sistema operativo si no lo tenemos instalado, este sistema operativo debe ser mayor a Windows 7 para que todo vaya bien y sin errores, después de esto debemos instalar un navegador que vendrá siendo nuestra base o soporte para poder visualizar el sistema de información o base de datos y por último tenemos que instalar el XAMPP o el WAMP cualquiera de los dos son muy buenos para la ejecución del sistema de información.

**Requisitos para el administrador y empleados (xampp para pruebas de manera local).**

* Adaptadores de Red.
* Acceso a internet.
* Procesador de 3.80GHz o superior.
* Memoria mínima de RAM 6Gb.
* Mouse.
* Teclado.
* Monitor.

**Requisitos para Clientes de la Empresa.**

* Adaptadores o chips de red.
* Acceso a internet.
* Procesador de 2.80GHz o superior.
* Memoria mínima de RAM 500Mb.
* Mouse (computador).
* Teclado (computador).
* Monitor o Display.

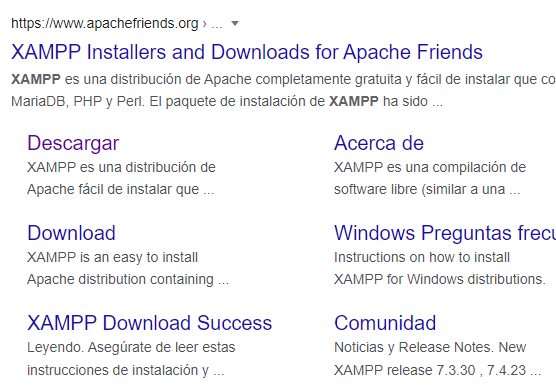
**Interfaces de software**

* Sistema operativo: NA (Cualquier persona).
* Explorador: NA (Cualquier persona).
* Excel (Para ver los reportes - Administradores y empleados).
* Xampp (Para pruebas de manera local - Administradores y empleados).
  1. **Detalles del proceso de instalación**

**XAMPP:**

Para este primer proceso de instalación buscaremos el programa para la descarga, seleccionamos el apartado.

<https://www.google.com/search?q=xampp&oq=&aqs=chrome.2.69i59i450l8.3469387j0j1&sourceid=chrome&ie=UTF-8>



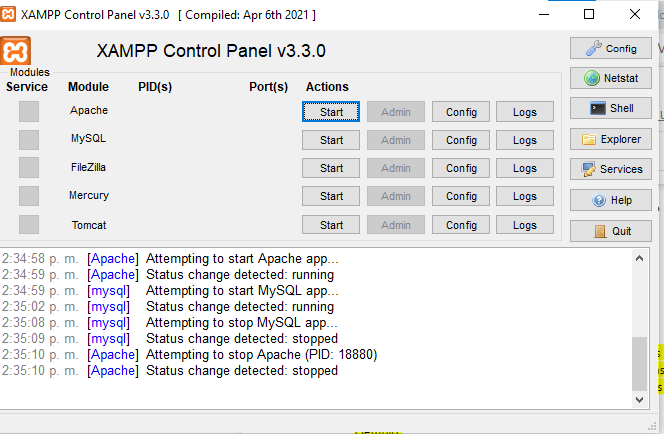
Seleccionamos el apartado descarga.

<https://www.apachefriends.org/es/download.html>

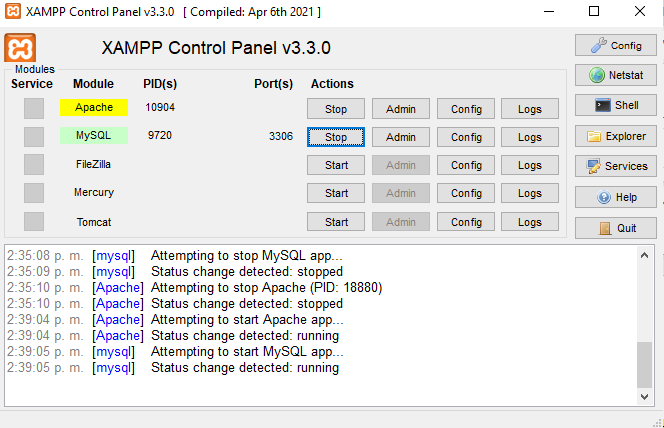


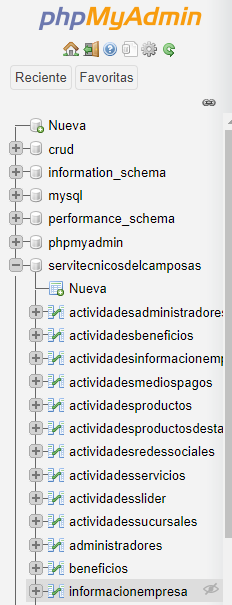
Se escoge la versión que desee instalar dependiendo del Windows y se procede a la instalación del programa para su uso.

<http://localhost/phpmyadmin/>



Una vez allí se selecciona el apache y el MySQL se les da Start a ambos para que quede activada la opción para visualizar la base de datos o el sistema de información.





y ya quedaría la aplicación descargada (XAMPP) para visualizar la base de datos y el sistema de información.

* 1. **Detalles de configuración de la aplicación**

En caso de usar XAMPP, no es necesario modificar la ruta de instalación ni ningún valor en el proceso.

**a. Variables de ambiente**

No es necesario un cambio en las variables de ambiente.

**b. Parámetros de aplicaciones**

En caso de trabajar con XAMPP, es necesario ir a la ubicación de instalación de xampp y encontrar el archivo llamado “config.inc.php” ubicado en “C:\xampp\phpMyAdmin\”. Al entrar ahí, es buscar alrededor de la línea 19 el código “$cfg['Servers'][$i]['auth\_type'] = 'config'; “ y cambiar la palabra “config” por “cookie”.

1. **Archivos de configuración**

En caso de usar XAMPP, para que el sistema de información funcione, es necesario mover la carpeta del proyecto a la siguiente ruta de la carpeta del xampp: “C:\xampp\htdocs\”. De esta manera, el sistema podrá trabajar en el servidor y base de datos local que brinda xampp.

1. **Lista de contactos técnicos**

| **Nombre completo** | **Empresa/Unidad Ejecutora** | **Módulo que atiende** | **Teléfonos/Beeper** |
| --- | --- | --- | --- |
| Victor Manuel Osorio Parra | Servicio Nacional de Aprendizaje SENA | Módulo Gestión de Cuenta Administrador  Módulo Gestión de Cuenta Empleado  Módulo Gestión de Iniciar Sesión | 3143428541 |
| Oscar Alejandro Acero Silva | Servicio Nacional de Aprendizaje SENA | Módulo Gestión de Productos  Módulo Gestión de Productos destacados  Módulo Gestión de Medios de Pago | 3224116812 |
| Andres Felipe Tellez Chavez | Servicio Nacional de Aprendizaje SENA | Módulo Gestión de información de la empresa  Módulo Gestión de Beneficios  Módulo Gestión de Sucursales | 3108110813 |
| Jhojan Danian Orjuela Barahona | Servicio Nacional de Aprendizaje SENA | Módulo Gestión de Servicios  Módulo Gestión de Redes sociales | 3022267286 |
| Angie Ximena Gonzalez Bermudez | Servicio Nacional de Aprendizaje SENA | Módulo Gestión del Slider  Módulo Gestión de Reportes | 3118870377 |

**10. Procesos de continuidad y contingencia**

Las funcionalidades propuestas en el sistema de información se proponen de acuerdo a las especificaciones de requerimientos del cliente, que podrá evidenciar en la Documento **NORMA** **IEEE 830 ( Norma IEEE 830 de 1998 Análisis, diseño, desarrollo e implantación de un sistema de información web para venta de maquinaria y servicios de la industria del Agro para la Compañía "Servitecnicos del campo SAS",docx)** del sistema ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA VENTAS DE MAQUINARIA Y SERVICIOS DE LA INDUSTRIA DEL AGRO PARA LA COMPAÑÍA "SERVITECNICOS DEL CAMPO S.A.S", garantizando todos los requisitos funcionales y no funcionales.

**11. Descripción de usuarios**

* 1. **Usuarios de base de datos**

**Nombre del usuario:** Nombre creado por el administrador en la base de datos.

**Descripción/Propósito::** Este usuario de base de datos, es el usuario universal del sistema de información, tendrá una credencial de acceso e integrará toda la arquitectura de la base de datos.

**Grupos a los que pertenece:** Administradores.

**Privilegios generales a nivel de base de datos, y privilegios sobre objetos de las mismas (tablas, vistas, relaciones, etc.):**

El usuario tendrá un acceso total a las tablas, vistas, procedimientos, tareas programadas y demás funcionalidades de la base de datos.  
  
**Nombre del usuario:** Nombre creado por el empleado por medio de la interfaz del administrador en la base de datos.

**Descripción/Propósito:** Este usuario de base de datos, es el usuario universal del sistema de información, tendrá una credencial de acceso e integrará toda la arquitectura de la base de datos.

**Grupos a los que pertenece:** Empleados.

**Privilegios generales a nivel de base de datos, y privilegios sobre objetos de las mismas (tablas, vistas, relaciones, etc.):**

El empleado tendrá acceso solamente a crear, actualizar y eliminar productos únicamente de la base de datos servitecnicosdelcamposas.

**Nombre del usuario:**  clienteservitecnicos.

**Descripción/Propósito:** Este usuario solo tendrá como función principal el leer los pedidos, es decir, solo podrá leer los productos que se encuentran en el sistema de información.

**Grupos a los que pertenece:** Clientes

**Privilegios dentro de la aplicación:** Tiene privilegios a los siguientes permisos:

* Leer