

Laureate International Universities



**LAUREATE**  
EDUCATION INC

*“Módulo de Alertas Para Renovación De Contratos”*

Análisis y Diseño del Sistema

Ing. En Sistemas:

- ◆ Cristian Javier Turcios

Tegucigalpa, M.D.C., 17 de Agosto de 2016

# Contenido

INTRODUCCIÓN	3
PROBLEMA DE LA INVESTIGACIÓN	4
JUSTIFICACIÓN	5
OBJETIVO GENERAL	6
OBJETIVOS ESPECÍFICOS	6
RECOLECCIÓN DE LA INFORMACIÓN	7
a) FACTIBILIDAD ECONÓMICA	8
b) FACTIBILIDAD TÉCNICA	9
c) FACTIBILIDAD OPERATIVA	11
ANÁLISIS E INTERPRETACIÓN DE LOS DATOS	12
REQUERIMIENTOS DEL SISTEMA	12
CASOS DE USO	14
FLUJOGRAMAS DEL SISTEMA	18
DESCRIPCIÓN DETALLADA DE LA BASE DE DATOS	19
Diseño de la Base de Datos	19
Diccionario de Datos	19
Descripción de las entidades de la base de datos	19
Diseño de la aplicación	21
CONCLUSIONES	24
BIBLIOGRAFÍA	25
ANEXOS	26
1. Acerca de Laureate	26
2. Casos de Uso	27
3. Flujogramas del sistema	29
4. Modelo Relacional	30
5. DICCIONARIO DE DATOS	Error! Bookmark not defined.
5. LENGUAJE DE DEFINICIÓN DE DATOS (DDL)	31
6. ENTREVISTA	Error! Bookmark not defined.
7. Diseño de Pantallas	82

## INTRODUCCIÓN

La planificación y exigencias de innovación se constituyen hoy en día elementos vitales para la sostenibilidad competitiva de una empresa. Las condiciones de un mercado abierto exigen a las empresas una preparación en donde el concepto de calidad deberá aplicarse en todas sus dimensiones.

Es por esto que el cambio es indispensable y debe ser planificado en base al análisis exhaustivo de la situación y de las condiciones en que opera la empresa. Para que esta innovación genere eficiencia y eficacia en los procesos que la empresa realiza, resulta imprescindible el uso de un sistema de información ya que el desarrollo tecnológico genera un nivel de calidad con el que la empresa obtiene competitividad en el mercado.

Por esta razón se desarrolla un estudio minucioso de la gestión de renovación de contratos, específicamente de la empresa “Laureate International Universities.”, ya que actualmente no cuenta con ningún sistema de cómputo propio que le permita llevar el registro y control total de los distintos contratos que ellos sostienen con distintas empresas las cuales le proveen servicios a dicha empresa; siendo este una de los procesos más importantes de dicha empresa, pero actualmente no se tiene un control ordenado de dicho proceso. Es por esta razón que el análisis se centra en este proceso.

Dando como resultado una solución para dicha problemática que se sintetiza en la opción de desarrollar e implementar un sistema de cómputo que persigue como objetivo principal la gestión de los contratos y alertas por correo electrónico a los distintos administradores o actores que se puedan encontrar en contacto con el sistema, esto ayudara a dar una disponibilidad de información verídica de una forma óptima y automatizada que garantice la calidad de las actividades realizadas en la empresa.

## PROBLEMA DE LA INVESTIGACIÓN

Laureate International Universities, filial de Laureate Education Inc., es una red de instituciones académicas privadas de educación superior de distintos países.

Reúne 76 instituciones de educación superior en 27 países del mundo, cerca de 600 000 estudiantes y más de 50 000 colaboradores en Europa, Latinoamérica, Norteamérica, Asia, África, Medio Oriente y Oceanía distribuidos en sus más de 100 campus, lo que la constituye como la mayor red mundial de universidades privadas.

La empresa actualmente no cuenta con un sistema automatizado que les ayude a agilizar sus procesos de renovación de contratos sostenidos con distintas empresas. Este hecho da como resultado que, debido a la gran cantidad de servicios, dominios y distintos contratos que ellos posean con gran cantidad de empresas, la recopilación de información, reportes y alertas de vencimiento se realice manualmente siendo un problema de costo y de tiempo para la empresa.

El hecho de que la información acerca de los distintos contratos que posee la empresa no este condensada en un solo sistema, además de ocasionar los problemas antes mencionados también complica la búsqueda de la fecha de vencimiento, duración del contrato, etc. Al momento de generar reportes o alertas preventivas de renovación, entre otros, ya que debido a la gran cantidad no se lleva un control eficiente.

## JUSTIFICACIÓN

La tecnología ha tenido un auge en los últimos años, esta da como resultado que el mundo tenga que adaptarse a este constante crecimiento tecnológico, lo cual significa que las empresas deben ser innovadores e implementar cambios en la manera que gestionan sus procesos, para que estos sean más eficientes y eficaces.

Si las empresas dejan de lado la tecnología, pronto quedaran enfrentadas a un mercado que no podrán satisfacer y donde la competencia podrá tener ventaja sobre ellas, donde se les considerara como una empresa con métodos rústicos incapaz de satisfacer las necesidades de sus clientes y sin posibilidades de crecimiento empresarial.

Es por ello que se pretende motivar a las empresas a que implementen software ya sea general o personalizado, en sus negocios; lo que les permitirá estar a la vanguardia tecnológica y poder expandir sus horizontes y visiones de expansión. Para ello los conocedores del campo informático y tecnológico debemos de proveer esas herramientas y soluciones, a través de sistemas de información que sean de fácil entendimiento y aplicación a sus negocios.

Un ejemplo de estas empresas es la empresa Laureate International Universities, una empresa que está en constante crecimiento tecnológico y que constantemente busca la automatización de todos sus procesos y se encuentra trabajando diario en la implementación de ellos, ya que de quedarse atrás sin la innovación los procesos serían más tardíos, el servicio y la entrega de información a sus clientes sería más retrasado.

Por lo tanto se persigue desarrollar un sistema que cumpla con una de las necesidades que ha surgido en la empresa, pero sobre todo que agilice y facilite las actividades que se desarrollan para que Laureate brinde un servicio eficiente, notando que una necesidad que ha surgido es un sistema donde se condense la información de todos los distintos contratos que la empresa posee y poder agilizar y condensar la información en un solo sistema y no tener que buscar la información de los contratos en cada una de las empresas que le presten servicio a Laureate.

## OBJETIVO GENERAL

- ♦ Realizar un análisis de los distintos contratos que posee la empresa, para así poder proponer una solución automatizada orientada a resolver los problemas encontrados.

## OBJETIVOS ESPECIFICOS

- ♦ Proponer un modelo que brinde la solución para agilizar los procesos en la empresa.
- ♦ Orientar el modulo propuesto hacia la escalabilidad de un sistema general a futuro.

## RECOLECCIÓN DE LA INFORMACIÓN

La recolección de la información se hizo mediante la aplicación de reuniones con los administradores que manejan los distintos contratos que maneja la empresa en donde ellos prepararon una serie de documentos donde se encontraba la información condensada de la variedad de contratos que ellos poseían y los elementos más significativos que tenía cada tipo de contrato.

Además, se utilizó como instrumento otro método para la recolección de la información en donde el analista junto con uno de los administradores visitó un sitio web ([freshservice.com](http://freshservice.com)) que poseía gran variedad de elementos que la empresa consideraba importante tener en un sistema propio personalizándolo con alguna de las características que ellos consideraban elementales para la empresa.

### *INSTRUMENTOS UTILIZADOS.*

#### ♦ **Reunión:**

Se realizó una entrevista el día xx de mayo de 2016 que duró aproximadamente 30 minutos. En donde se pudo conversar con uno de los administradores de los contratos que maneja la empresa, quien contestó gustosa a todas las preguntas que se le realizaron y además proporciono documentos donde se encontraba información de los documentos y distintos tipos de contratos que posee la empresa con las empresas que le brindan algún servicio, la cual será presentada en los anexos del presente informe.

#### ♦ **Observación:**

Al realizar el análisis del sistema, además de realizar las reuniones se dio una visita con uno de los administradores al sitio en internet [freshservice.com](http://freshservice.com) para ver la funcionalidad que ofrece este sistema web y poder incorporar alguno de los elementos que posee en el sistema que se piensa desarrollar para la empresa.

La utilización de estos instrumentos y métodos de recolección de información facilitan la obtención de la información, que después de ser estudiada y analizada ayudarán a poder brindar una solución factible en cuanto al análisis y diseño del sistema para la empresa.

## *ESTUDIO DE LA FACTIBILIDAD*

Al haber definido la problemática antes mencionada y establecer las necesidades de un sistema de información es necesario determinar qué tan factible podría ser éste para la empresa. Así también como la capacidad técnica e infraestructura que se requerirá en la implantación del sistema, además de mencionar los costos y los beneficios que el sistema traería.

A través del análisis se hizo posible generar un diseño del sistema propuesto y para ello se tomaron en cuenta algunos aspectos que se mencionan a continuación:

### **a) FACTIBILIDAD ECONOMICA**

Es importante considerar como aspecto fundamental la factibilidad económica, ya que no tendría sentido que una empresa invierta en un sistema que solo generará costos sin muchos beneficios.

Por ello se presentará un análisis costo-beneficio para determinar los costos que se generarán en la implantación del sistema propuesto así también los beneficios que trae consigo.

#### **♦ Análisis costo-beneficio.**

Como actualmente la empresa cuenta con varios sistemas de información se pretende mostrar los costos que implicaría un sistema contra los beneficios que traería.

El que la empresa actualmente cuente con varios sistemas informáticos podría disminuir los costos casi a 0 en cuanto a la puesta en marcha de un sistema, ya que se requiere cierto equipo con el que la empresa ya cuenta.

A continuación, se detallan los costos:

**Costos generales:** Estos costos son básicamente los que se generan a diario como ser material de oficina, accesorios, impresoras y en general todo aquello que se requiere para las tareas cotidianas que se realizan en la empresa.

**Costos de personal:** Estos son los costos que se generan por recursos humanos, es decir el personal que labora en la empresa y que le dará uso al sistema a implantar.



**Costos de hardware y software:** Debido a que la empresa cuenta con todo el equipo técnico necesario para el desarrollo e implantación del sistema no es necesario adquirir equipo nuevo para que el sistema pueda operar.

#### ♦ **Beneficios**

Después de mencionar los costos se mostrarán algunos de los beneficios que trae el sistema de información.

Beneficios Tangibles:

- ♦ Reducción de costos en papelería.
- ♦ Beneficios Intangibles:
- ♦ Optimización y agilización de las actividades dentro de la empresa, especialmente en el área de condensación de la información de los contratos.
- ♦ Facilidad en el manejo de la información.
- ♦ Disponibilidad de la información de forma rápida.
- ♦ Facilidad de búsqueda y actualización de la información.
- ♦ Buen aprovechamiento de los recursos tecnológicos instalados.
- ♦ Registro automatizado de la información.
- ♦ Seguridad de la información registrada.

#### **b) FACTIBILIDAD TÉCNICA**

Mediante la factibilidad técnica se logra determinar qué componentes técnicos se requieren en base al equipo con el que actualmente posee la empresa.

Los enfoques a tomar en cuenta para determinar esta factibilidad son: hardware y software.

### Hardware:

En cuanto a hardware se requiere:

- Un servidor.
- Computadoras personales:
  - Procesador Intel Pentium 4 o superior
  - Tarjeta Madre.
  - 1 GB de Memoria RAM.
  - Disco Duro 100 MB de espacio libre.
  - Tarjeta de Red.
  - Tarjeta de Video.
  - Monitor SVGA.
  - Teclado.
  - Mouse.
- Una red local (intranet):
  - Router.
  - Switch.
  - Cableado estructurado (UTP).
  - Conectores (RJ-45).
- Regulador de voltaje.
- Impresora.
- UPS.

**Hardware disponible:**

La empresa cuenta con todo el equipo mencionado anteriormente, equipo de nueva generación para lo cual la implementación de un nuevo sistema no generaría costo extra a la empresa.

**Software:**

La empresa no posee ninguna clase de software propio de la empresa para el manejo del proceso de condensación de información de contratos, más que el software que le ofrece algunas empresas como GoDaddy.com que ofrece un mini sistema de administración de dominios, pero la empresa al contar con contratos de otras empresas ocupa un sistema donde se almacene la información no solo de GoDaddy.com sino de todas las demás compañías con quien tiene contratos, como xoom, Skype, Microsoft, etc.

**C) FACTIBILIDAD OPERATIVA**

La factibilidad operativa se refiere a que debe existir el personal capacitado requerido para llevar a cabo el proyecto, y así mismo deben existir los usuarios dispuestos a emplear y manejar el sistema.

Es decir, el personal debe estar a la disponibilidad de utilizar el sistema, para que éste logre proporcionar todos los beneficios y se vean reflejados en las actividades de la empresa.

La empresa cuenta con dicho personal capacitado en el área tecnológica y dispuesto a la utilización del sistema por lo cual tampoco se incurriría en gastos extras como capacitaciones para que el personal pueda aprender a utilizar un sistema web como el que se piensa desarrollar.

## ANÁLISIS E INTERPRETACIÓN DE LOS DATOS

### *REQUERIMIENTOS DEL SISTEMA*

El sistema debe satisfacer algunos requerimientos que ayudaran a la empresa a tener un mejor control de la facturación de la empresa.

- 1) Registro de nuevos contratos.
- 2) Almacenamiento de los registros de contratos ya existentes.
- 3) Alertas de la fecha de vencimiento con anticipación de los contratos ya existentes.
- 4) Registros de proveedores y servicios que ofrecen.
- 5) Generación de reportes.

#### **1) Registro de nuevos contratos**

Este requerimiento es necesario ya que así es como se podrá ofrecer una automatización del problema que presenta la empresa ya que por este medio es como se podrá lograr la condensación de toda la información de los contratos en un sistema único y personalizado para la empresa.

#### **2) Almacenamiento de los registros de contratos ya existentes**

Es necesario almacenar la información de los distintos contratos que la empresa ya posee y así poder lograr una automatización al 100% para que el sistema no solo ofrezca las ventajas que tendrá a los nuevos contratos sino también a los existentes, para así poder generar alertas de renovación o generar reportes específicos según los requerimientos de los administradores.

#### **3) Alertas de la fecha de vencimiento con anticipación de los contratos ya existentes**

Este requerimiento es necesario debido a que el problema principal es que no tienen un control eficiente de la fecha de vencimiento de los distintos servicios que tienen con las distintas empresas, esto debido a la gran cantidad de servicios que la empresa Laureate tiene contratados y por lo cual es complicado el manejo de tanta información y fechas de

vencimiento de los servicios para los administradores debido a que el flujo de información de estos elementos es bastante grande.

Con la implementación de alertas próximas a las fechas de vencimiento de los distintos servicios que posee la empresa los administradores podrán hacer todas las gestiones necesarias con tiempo para poder renovar los servicios y no manejar las fechas de vencimiento ellos, ya con recordatorios en calendario o cualquier otro medio que se pueda utilizar, sino que el propio sistema genere estas alertas y brinde una mayor facilidad y ventaja que ayudara a una mejor administración de los servicios que la empresa tenga contratados.

#### **4) Registros de Proveedores y servicios que ofrecen**

Es necesario llevar un registro de los distintos proveedores de servicio que la empresa tiene, así como los distintos servicios que cada uno de estos proveedores, esto facilitara y agilizara el proceso para futuras contrataciones de nuevos servicios que desee contratar la empresa, no será necesario buscar por la web una empresa que provee algún servicio específico, ya que el sistema que se pretende desarrollar contara con dicha información en sus registros

#### **5) Generación de Reportes**

Toda la información de contratos, proveedores, lista de servicios que ofrezca un proveedor, alertas de contratos próximos a vencer, o contratos activos nos ayudaran a generar reportes mensuales y anuales según los requiera necesario los administradores lo que les facilitara llevar un mejor control en estas actividades.

## CASOS DE USO

### Identificación de Actores

Se identificaron los siguientes actores en el sistema:

Actor	Tipo	Rol
Gerente	Primario	Administrador del sistema
Administrador Contratos	Primario	Administrador del sistema
Usuario	Secundario	Visualización del sistema, Generar Reportes

### Casos de uso

Los casos de uso a implementar en el sistema serán los siguientes:

- 1) Ingreso al sistema
- 2) Registrar Proveedor
- 3) Registrar Producto
- 4) Registrar nuevo contrato
- 5) Buscar Proveedor
- 6) Buscar Producto
- 7) Buscar Contrato
- 8) Modificar Proveedor, Producto, Contrato
- 9) Eliminar Proveedor, Producto Contrato
- 10) Generar Reportes

Nota: Para los casos de uso se asume que ya existe un usuario creado en el sistema

### 1) Caso de uso: ingreso al Sistema

Actores	Gerente, Administrador, Usuario
<b>Objetivos</b>	Tener acceso al sistema.
<b>Pre-condiciones</b>	Para tener acceso al sistema se deberá estar previamente registrado con un nombre de usuario y una contraseña.
<b>Flujo Básico</b>	Se ingresa el nombre de usuario y contraseña para verificar la existencia de los mismos en la base de datos del sistema y posteriormente se habilitarán las pantallas a las que el usuario tendrá permitido acceso.
<b>Flujos Alternativos</b>	En caso de que el usuario haya olvidado su contraseña, se podrá recuperar a través de preguntas de seguridad.

### 2) Caso de Uso: Registrar Proveedor

Actores	Gerente, Administrador
<b>Objetivos</b>	Registrar un proveedor en el sistema.
<b>Pre-condiciones</b>	Que el proveedor preste un servicio a la empresa.
<b>Flujo Básico</b>	Se contratan los servicios de un proveedor ya sea vía telefónica o vía internet, una vez contratados los servicios, el gerente o el administrador del sistema ingresará los datos del proveedor en el sistema como nombre del proveedor, correo, contacto, ubicación etc.
<b>Flujos Alternativos</b>	En caso de no encontrarse presente el gerente, el contador podrá ingresar los datos del cliente al sistema, los cuales ya estarán previamente proporcionados por el gerente.

### 3) Caso de Uso: Registrar nuevo contrato

Actores	Administrador
<b>Objetivos</b>	Registrar un nuevo contrato en el sistema.
<b>Pre-condiciones</b>	Que el contrato se encuentre activo y se tenga toda la información necesaria para poder ingresar la información al sistema.
<b>Flujo Básico</b>	El administrador del sistema digitalizara toda la información que posea el contrato en los campos que se proveen en el sistema como proveedor del servicio, nombre contrato, fecha expiración, etc. una vez se hayan llenado todos los campos requeridos en el sistema, el administrador guardara el contrato y este ya quedara disponible para su visualización dentro del sistema,
<b>Flujos Alternativos</b>	Si un proveedor no está registrado en la base de datos, el administrador deberá ingresarlo.

### 4) Caso de Uso: Buscar Proveedor

Actores	Administrador, usuario
<b>Objetivos</b>	Verificar y buscar un proveedor en el sistema.
<b>Pre-condiciones</b>	Estar logueado en el sistema.
<b>Flujo Básico</b>	El usuario o administrador tendrá la posibilidad de ingresar el nombre de un proveedor para que el sistema verifique la existencia de posibles coincidencias y despliegue los datos detallados del proveedor si este se encuentra registrado en el sistema.
<b>Flujos Alternativos</b>	De no ser encontrado y ser requerido el administrador deberá registrar el proveedor en el sistema.

### 5) Caso de Uso: Buscar Contrato

Actores	Administrador, Usuario
<b>Objetivos</b>	Mostrar el detalle de un contrato específico en el sistema.
<b>Pre-condiciones</b>	Que los contratos hayan sido previamente almacenados en el sistema.
<b>Flujo Básico</b>	El usuario o administrador tiene la posibilidad de buscar un contrato introduciendo diferentes tipos de condiciones como ser: número de contrato, nombre del contrato, la fecha en que se generó el contrato, fecha finalización del contrato, generándose a detalle los elementos contenidos en el contrato
<b>Flujos Alternativos</b>	De no ser encontrado y ser requerido el administrador deberá registrar el contrato en el sistema.



## 6) Caso de Uso: Modificar Proveedor, Contrato

Actores	Administrador
<b>Objetivos</b>	Realizar cualquier cambio en los registros de un contrato o un proveedor.
<b>Pre-condiciones</b>	Que el contrato o el proveedor haya sido creado en el sistema.
<b>Flujo Básico</b>	El usuario pedirá al sistema modificar un contrato o un proveedor, esto en caso de alguna irregularidad que se haya presentado en la introducción de datos iniciales.
<b>Flujos Alternativos</b>	Ninguno

## 7) Caso de Uso: Eliminar Proveedor, Contrato

Actores	Administrador
<b>Objetivos</b>	Realizar la eliminación en los registros de un contrato o un proveedor.
<b>Pre-condiciones</b>	Que el contrato o el proveedor haya sido creado en el sistema.
<b>Flujo Básico</b>	El usuario pedirá al sistema modificar un contrato o un proveedor, esto en caso de alguna irregularidad que se haya presentado en la introducción de datos iniciales.
<b>Flujos Alternativos</b>	Ninguno

## 8) Caso de Uso: Generar Reportes

Actores	Usuario, Administrador
<b>Objetivos</b>	Realizar reportes personalizados según lo requiera el usuario, como ser un reporte de los contratos que estén ingresados en el sistema, reporte de los contratos próximos a vencer, reportes de todos los proveedores, etc.
<b>Pre-condiciones</b>	Que los datos de los contratos y proveedores hayan sido ingresados para que estén almacenados en la base de datos del sistema.
<b>Flujo Básico</b>	El usuario pedirá al sistema generar un reporte, el sistema de información pedirá las especificaciones de qué tipo de reportes desea, el usuario deberá de marcar la información que desee conocer en el reporte. El sistema le imprimirá en pantalla el reporte y le dará la opción de imprimirlo en papel también.
<b>Flujos Alternativos</b>	De no encontrar en el sistema la información para generar el reporte el sistema le mostrará un mensaje diciendo que no se podrá realizar el reporte pedido, ya que no existe la información solicitada.

Diagramas de Caso de Uso: **Anexo 2**

### *FLUJOGRAMAS DEL SISTEMA*

Se presentan los flujogramas que tendrá el sistema.

Ver: **Anexo 3**

## DESCRIPCIÓN DETALLADA DE LA BASE DE DATOS

### *DISEÑO DE LA BASE DE DATOS*

Se realizó el modelo relacional de la base de datos, donde se presenta la colección de relaciones, tuplas, atributos y dominios, que nos ayudaran a gestionar el flujo de información del sistema.

Ver: **Anexo 4**

### *DICCIONARIO DE DATOS*

Se facilita el diccionario de datos contiene las características lógicas de los datos que se van a utilizar en el sistema. Se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema.

Ver: **Anexo 5**

### *DESCRIPCIÓN DE LAS ENTIDADES DE LA BASE DE DATOS*

1. **Área Contratos:** En esta tabla se hace la distinción a la que un área de contrato permacera, hasta el momento de la redacción de este documento las áreas existentes son: Main Office, It Support Corporate. Es importante que un contrato permanezca a un área ya que así se podrá hacer un filtrado de los contratos que se deben mostrar a los usuarios según el área al que estos pertenezcan.
2. **Borrador Contratos:** Esta tabla es una copia casi exacta de la tabla contratos (que se detallara a continuación), el objetivo de esta tabla es poder permitir al usuario guardar como un contrato como borrador y luego poder seguir editándolo
3. **Contratos:** Esta es la tabla principal del sistema, aquí es donde se guardan los contratos, con todas sus llaves foráneas hacia las otras tablas.
4. **Estados:** En esta tabla se detalla el estado que un contrato pueda tener, los tipos de estados que un contrato puede tener hasta el momento es 1. Activo: este estado se da cuenta la fecha de pago no está cerca y la fecha de vencimiento del contrato aún no se acerca, 2. Próximo a vencer: este estado se da cuando la fecha de pago esta próxima o la fecha de

vencimiento del contrato se encuentra cerca 3. Vencido: este estado se da cuando un contrato ya ha vencido.

5. **Método Pagos:** En esta tabla se especifica los distintos métodos de pagos que un contrato puede tener, actualmente los distintos métodos de pago que existen es: cheque, efectivo, tarjeta, transferencia bancaria.
6. **Orden Compras:** Esta tabla se refiere a las órdenes de compra previas que pueden generarse para un contrato, por ejemplo, antes de firmar un contrato con una empresa se genera una orden de compra y luego se firma un contrato, estas órdenes de compra deben ser registradas en el sistema también y se registran en esta tabla
7. **Perfiles:** Esta tabla es donde se registran los distintos perfiles que se presentan en el sistema, estos perfiles tienen ciertas restricciones, por ejemplo, las acciones que puedan realizar, los tipos de contratos que puedan ver, etc.
8. **Productos:** En esta tabla se detalla los distintos productos que un contrato puede tener, estas tablas también tienen una relación directa con proveedores, ya que para registrar un producto se debe especificar cuál es el proveedor que distribuye y manufactura los productos
9. **Proveedores:** Esta tabla sirve para detallar los proveedores de los productos que posee un contrato, por ejemplo, un proveedor podría ser google, Amazon, etc.
10. **Recordatorios:** tabla que se usa para almacenar las fechas anticipadas de recordatorios de un contrato, por ejemplo, se puede registrar un contrato, pero se desea que 3, 7, 15, o 31 días antes se envíe una alerta por correo a un usuario, aquí es donde se almacena los días previos a que se envíen estas alertas contiene los clientes existentes en la empresa.
11. **Tarjetas:** Aquí es donde se almacena la distinta información de las tarjetas con las cuales se puede pagar un contrato.
12. **Tipo Contratos:** el objetivo de esta tabla es diferenciar los distintos tipos de contrato que puedan existir, por ejemplo, un contrato puede ser de arrendamiento, como la renta del piso de un edificio, licencia, para la licencia de un software, o de mantenimiento por ejemplo para alguna que brinde algún soporte técnico.
13. **Tipo pagos:** esta tabla especifica los distintos periodos de pago que un contrato pueda tener, hasta el momento los tipos de pagos registrados son pagos de contratos mensuales, trimestrales, cuatrimestrales y anuales.

- 14. **Tipo proveedores:** esta tabla diferencia los distintos tipos de proveedores que puedan existir, se hace uso de esta tabla ya que un proveedor puede ser manufacturero (el que fabrica el producto por ejemplo Dell) y proveedor distribuidor (el que vende y distribuye el producto por ejemplo JetStereo)
- 15. **Tipo Tarjetas:** En esta tabla se guardan los distintos tipos de tarjetas, por ejemplo, Visa, MasterCard, American Express y Discovery.
- 16. **Usuarios:** Esta tabla sirve para guardar el registro de los distintos usuarios que existen en el sistema, en esta tabla se hace una relación a la tabla de perfiles, ya que un usuario para poder registrarse en un sistema, se debe registrar con un perfil en específico.
- 17. **Usuarios\_Perfiles\_Permisos:** Esta tabla es la principal donde el sistema verifica si un usuario está autorizado a realizar cierta acción o no según el perfil que este posee (El campo permiso no se utiliza para nada, por eso ni siquiera se detalló el significado de la tabla permiso)

## Diseño de la aplicación

### *PLANTEAMIENTO PARA EL DISEÑO DE INTERFACES*

#### **¿Quiénes son los usuarios del sistema?**

Principalmente existirán dos usuarios primarios que son los que interactúan de manera directa con el sistema de alertas de renovación de contratos de la empresa “Lauréate International Universities”. Uno de ellos es el gerente quien es el encargado de supervisar el trabajo de todos los demás colaboradores de la empresa, por tal razón podrá realizar dentro del sistema actividades de control como ser brindar los usuarios, agregar y modificar contratos.

El otro usuario que tendrá interacción de manera directa con el sistema es el administrador quien es el encargado de llevar el estado y la renovación de los contratos, generación de reportes y agregar nuevos usuarios en caso de ser necesario.

#### **¿Cómo aprende el usuario a interactuar con el nuevo sistema de cómputo?**

Debido a la importancia de minimizar tiempo para los usuarios del sistema se tratará que el sistema sea lo más intuitivo posible y de fácil manejo, aparte los usuarios que manejan son expertos en el área tecnológica por lo cual la curva de aprendizaje del sistema se vuelve realmente corta.

### **¿Qué desea el usuario?**

Actualmente el usuario maneja muchos contratos con distintas empresas, como ser dominios, licencias y otros tipos de contratos los cuales tienen una fecha de vencimiento límite, el usuario desea un sistema que recuerde con anticipación por medio de un correo electrónico el vencimiento de una licencia, dominio, etc., y poder anticiparse a que no se cancele el contrato por falta de renovación. El objetivo del sistema será enfocarse en esto, introducir toda la información de los distintos contratos que la empresa ya posee y poder desarrollar el sistema de alertas por correo para que el administrador del sistema tenga esa facilidad de tener un recordatorio de renovación de todos los distintos contratos en un mismo sistema.

## ***DISEÑO DE LA INTERFAZ DE USUARIO***

Una interface simple y eficiente para la interacción del usuario con el sistema es de vital importancia ya que el ahorro de tiempo y evitar capacitaciones es indispensable para la empresa, es por tal razón que el diseño está centrado en el usuario considerando lo siguiente:

- ◆ Producción de mensajes de alerta y de error significativos para el usuario, como ser la entrada errónea de algún dato.
- ◆ Proporción al usuario de información visual para asegurarse de que la comunicación con la aplicación está bien establecida.
- ◆ Mostrar información que sea relevante dentro del contexto a manera de ayuda para que sea más cómodo el manejo y la interacción con el sistema.

## ***DISEÑO DE LA ENTRADA DE DATOS***

El diseño de la entrada consiste en el desarrollo de especificaciones y procedimientos para la preparación de datos, la realización de pasos necesarios para preparar los datos de una transacción en una forma utilizable para su procesamiento y posteriormente sirva como reporte, para el análisis de las entradas se tomó en cuenta lo siguiente:

- ◆ Los datos que ingresan al sistema deben de estar organizados de manera consistente, evitando la redundancia es por ello que se brinda métodos de entrada restringidos y de fácil ingreso.
- ◆ Los medios utilizados por el sistema: que para este caso será un teclado digital o analógico por la facilidad de ingreso de los datos.
- ◆ La forma en que se debe disponer o codificar los datos, se utiliza una base de datos relacional de manera que en la captura de los datos se almacenan en las direcciones correspondientes.
- ◆ Los controles que servirán de guía a los usuarios para la entrada de datos, que en su mayoría serán controles de selección para los datos que así lo requieran.
- ◆ Validación necesaria de datos y transacciones que se controla por medio de la restricción de dominio para cada tipo de dato y así detectar y controlar errores.

### *DISEÑO DE SALIDA DE DATOS*

El diseño de la salida de datos consiste en el desarrollo de los resultados generados por el sistema, estos pueden ser mensajes, reportes y documentos, estos deben realizar lo siguiente. El sistema brindara solución a uno de los principales inconvenientes del proceso de facturación de la empresa que es la generación de reportes, mensuales, anuales, filtrados por proveedores, contratos, fechas, la forma de presentación será por medio de pantalla con la opción de exportación a tipos de dato como ser hojas de cálculo, documento .pdf o imágenes para ser enviadas por correo electrónico.

### *DISEÑO DE PANTALLAS*

Ver: **Anexo 7**

## CONCLUSIONES

- ◆ A través del análisis realizado de las condiciones de la empresa determino que un sistema de cómputo hecho a la medida dará solución completa para dicha problemática.
- ◆ La implementación de esta aplicación no solo ayudará a la condensación de la información de los distintos contratos que posee la empresa, si no que permitirá la fácil administración de la información de los contratos, proveedores y las alertas sobre las fechas de vencimiento de cada uno de ellos, teniendo así una mejor gestión en esta tarea.
- ◆ Dicho módulo desarrollado tiene la ventaja de que puede acoplarse a distintos módulos desarrollados en el futuro ya que puede utilizarse por un amplio periodo de tiempo puesto que satisface con todas las necesidades definidas en el análisis del problema.
- ◆ El beneficio que los reportes generados por la aplicación ayudaran a la toma de decisiones ya que brindaran la información de los servicios realizados por la empresa en cierto periodo de tiempo, brindando un panorama de la situación actual de la empresa.



## BIBLIOGRAFIA

- ◆ Fundamentos.de.Bases.de.datos.Abraham.Silberschatz.McGRAW-HILL

## ANEXOS

### 1. ACERCA DE LAUREATE INTERNATIONAL UNIVERSITIES



#### **Mission**

Expanding access to quality higher education to make the world a better place

#### **Description**

Throughout our worldwide network of higher education institutions, we share a mission to make quality higher education accessible and affordable so more students can pursue their dreams. It's a mission we believe will help make the world a better place.

Laureate's network institutions deliver professional-oriented programs in a wide range of disciplines that generate strong interest from students and provide attractive employment outcomes.

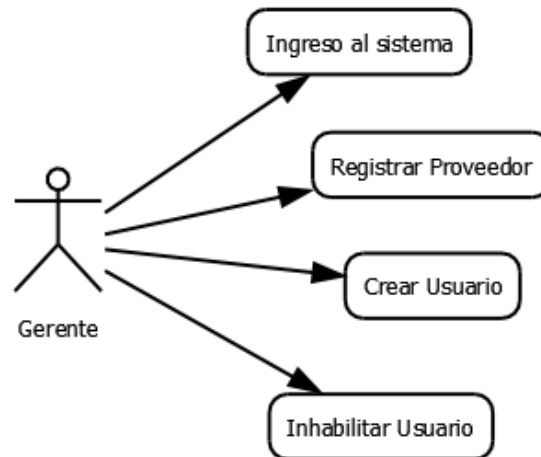
We believe in the power of education to transform lives and remain committed to making a positive, enduring impact in the communities we serve. When our students succeed, countries prosper and societies benefit.

## 1. CASOS DE USO



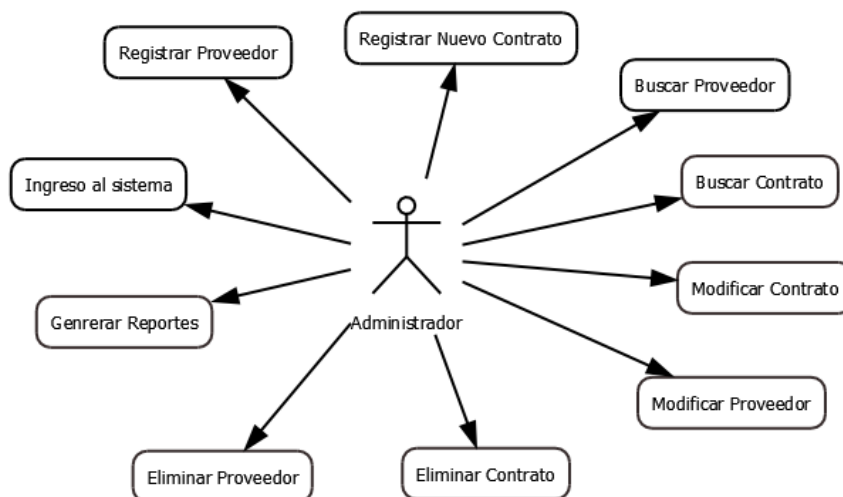
# LAUREATE

EDUCATION INC<sup>®</sup>



# LAUREATE

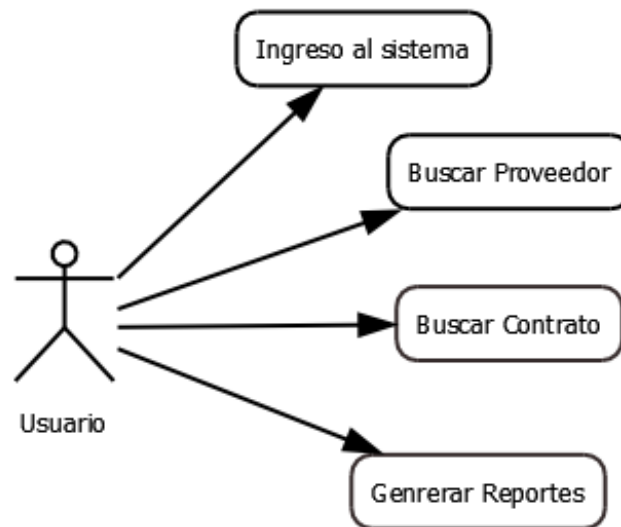
EDUCATION INC<sup>®</sup>



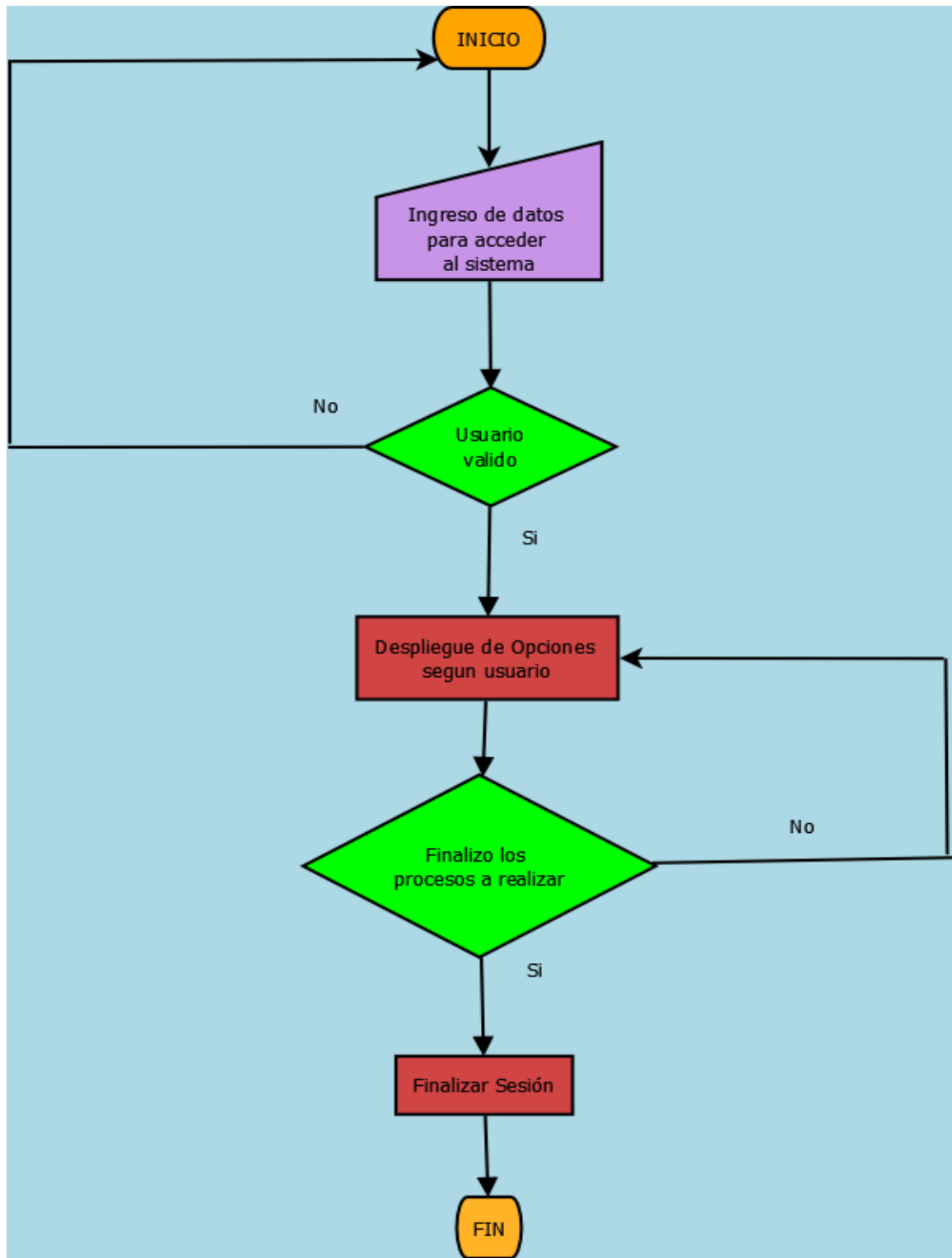


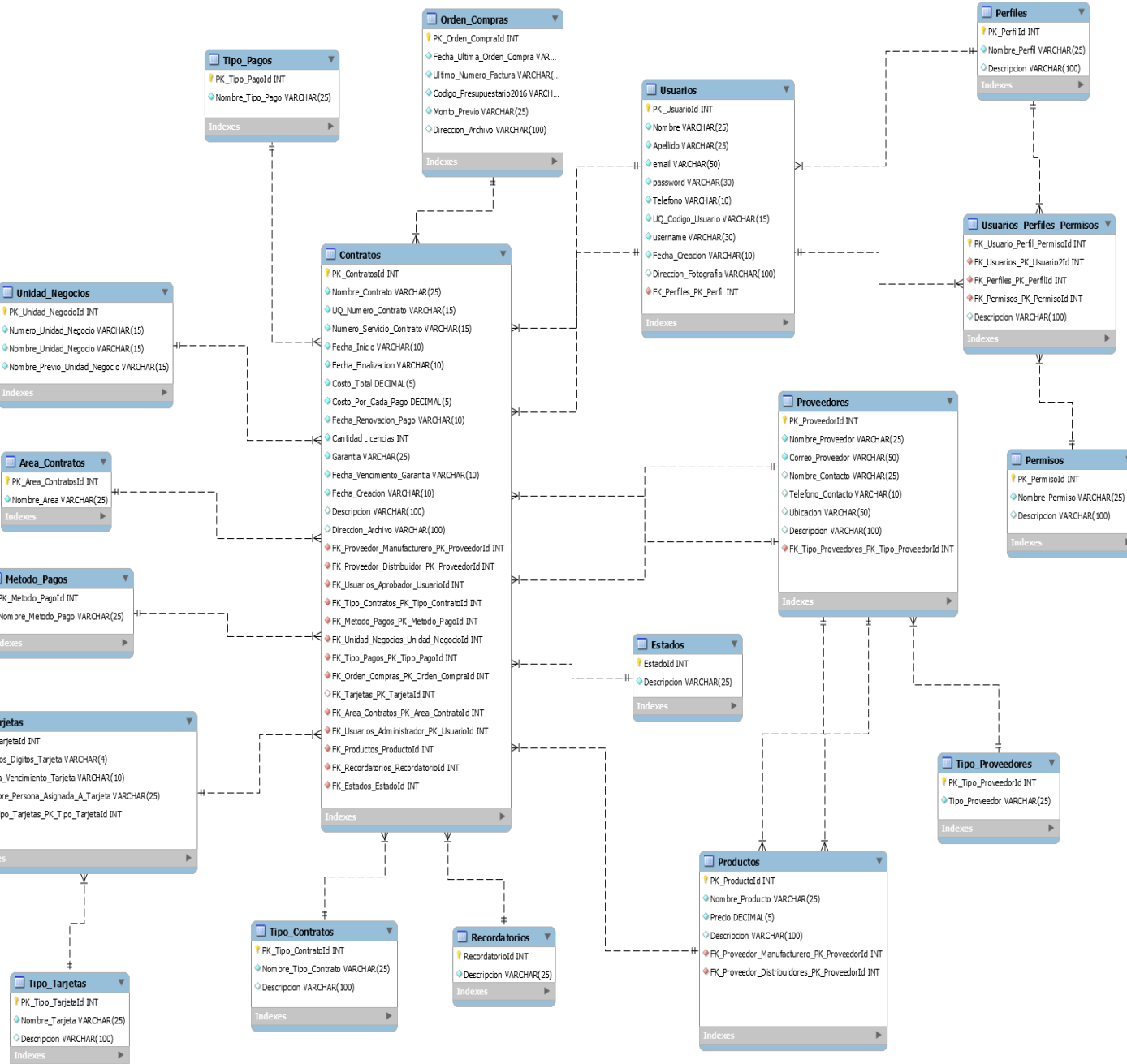
# LAUREATE

## EDUCATION INC<sup>®</sup>



## 2. FLUJOGRAMAS DEL SISTEMA





## 5. LENGUAJE DE DEFINICIÓN DE DATOS (DDL)

```

USE [master]
GO
/***** Object: Database [sistema_alertas]    Script Date: 7/20/2016 2:01:57 PM *****/
CREATE DATABASE [sistema_alertas]
    CONTAINMENT = NONE
    ON PRIMARY
( NAME = N'sistema_alertas', FILENAME = N'G:\SQLDATA\MSSQLSERVER\sistema_alertas.mdf' , SIZE =
4096KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
LOG ON
( NAME = N'sistema_alertas_log', FILENAME = N'H:\SQLLOGS\MSSQLSERVER\sistema_alertas_log.ldf' , SIZE
= 1024KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)
GO
ALTER DATABASE [sistema_alertas] SET COMPATIBILITY_LEVEL = 120
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [sistema_alertas].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [sistema_alertas] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [sistema_alertas] SET ANSI_NULLS OFF
GO
ALTER DATABASE [sistema_alertas] SET ANSI_PADDING OFF
GO
ALTER DATABASE [sistema_alertas] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [sistema_alertas] SET ARITHABORT OFF
GO
ALTER DATABASE [sistema_alertas] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [sistema_alertas] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [sistema_alertas] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [sistema_alertas] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [sistema_alertas] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [sistema_alertas] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [sistema_alertas] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [sistema_alertas] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [sistema_alertas] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [sistema_alertas] SET DISABLE_BROKER
GO
ALTER DATABASE [sistema_alertas] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO

```

```

ALTER DATABASE [sistema_alertas] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [sistema_alertas] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [sistema_alertas] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [sistema_alertas] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [sistema_alertas] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [sistema_alertas] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [sistema_alertas] SET RECOVERY SIMPLE
GO
ALTER DATABASE [sistema_alertas] SET MULTI_USER
GO
ALTER DATABASE [sistema_alertas] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [sistema_alertas] SET DB_CHAINING OFF
GO
ALTER DATABASE [sistema_alertas] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [sistema_alertas] SET TARGET_RECOVERY_TIME = 0 SECONDS
GO
ALTER DATABASE [sistema_alertas] SET DELAYED_DURABILITY = DISABLED
GO
EXEC sys.sp_db_vardecimal_storage_format N'sistema_alertas', N'ON'
GO
USE [sistema_alertas]
GO
/***** Object: User [HNSCNET\cturcios]    Script Date: 7/20/2016 2:01:57 PM *****/
CREATE USER [HNSCNET\cturcios] FOR LOGIN [HNSCNET\cturcios] WITH DEFAULT_SCHEMA=[dbo]
GO
/***** Object: User [contracker]    Script Date: 7/20/2016 2:01:57 PM *****/
CREATE USER [contracker] FOR LOGIN [contracker] WITH DEFAULT_SCHEMA=[dbo]
GO
ALTER ROLE [db_owner] ADD MEMBER [HNSCNET\cturcios]
GO
ALTER ROLE [db_owner] ADD MEMBER [contracker]
GO
/***** Object: Table [dbo].[area_contratos]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[area_contratos](
    [Area_ContratoId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre_Area] [varchar](25) NOT NULL,
    CONSTRAINT [PK_Area_ContratosId] PRIMARY KEY CLUSTERED
(
    [Area_ContratoId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

```



```

SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Borradorcontratos]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Borradorcontratos](
    [ContratoId2] [int] IDENTITY(1,1) NOT NULL,
    [Nombre_Contrato2] [varchar](25) NULL,
    [Numero_Contrato2] [varchar](15) NULL,
    [Numero_Servicio_Contrato2] [varchar](15) NULL,
    [Fecha_Inicio2] [date] NOT NULL,
    [Fecha_Finalizacion2] [date] NOT NULL,
    [Costo_Total2] [numeric](18, 2) NULL,
    [Costo_Por_Cada_Pago2] [numeric](18, 2) NULL,
    [Fecha_Renovacion_Pago2] [date] NOT NULL,
    [Cantidad_Licencias2] [int] NULL,
    [Garantia2] [varchar](25) NULL,
    [Fecha_Vencimiento_Garantia2] [date] NOT NULL,
    [Fecha_Creacion2] [date] NULL,
    [Descripcion2] [varchar](500) NULL,
    [Direccion_Archivo2] [varchar](100) NULL,
    [Proveedor_ManufactureroId2] [int] NULL,
    [Proveedor_DistribuidorId2] [int] NULL,
    [Usuario_AprobadorId2] [int] NULL,
    [Usuario_AdministradorId2] [int] NULL,
    [Tipo_ContratoId2] [int] NULL,
    [Metodo_PagoId2] [int] NULL,
    [Unidad_NegocioId2] [int] NULL,
    [Tipo_PagoId2] [int] NULL,
    [Orden_CompraId2] [int] NULL,
    [TarjetaId2] [int] NULL,
    [Area_ContratoId2] [int] NULL,
    [ProductoId2] [int] NULL,
    [RecordatorioId2] [int] NULL,
    [EstadoId2] [int] NULL,
    [Detalles_Producto2] [varchar](max) NULL,
    CONSTRAINT [PK_ContratoId2] PRIMARY KEY CLUSTERED
(
    [ContratoId2] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[contratos]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[contratos](

```

```

[ContratoId] [int] IDENTITY(1,1) NOT NULL,
[Nombre_Contrato] [varchar](25) NOT NULL,
[Numero_Contrato] [varchar](15) NOT NULL,
[Numero_Servicio_Contrato] [varchar](15) NOT NULL,
[Fecha_Inicio] [date] NOT NULL,
[Fecha_Finalizacion] [date] NOT NULL,
[Costo_Total] [numeric](18, 3) NOT NULL,
[Costo_Por_Cada_Pago] [numeric](18, 3) NOT NULL,
[Fecha_Renovacion_Pago] [date] NOT NULL,
[Cantidad_Licencias] [int] NULL,
[Garantia] [varchar](25) NULL,
[Fecha_Vencimiento_Garantia] [date] NOT NULL,
[Fecha_Creacion] [date] NOT NULL,
[Descripcion] [varchar](500) NULL DEFAULT (NULL),
[Direccion_Archivo] [varchar](100) NULL DEFAULT (NULL),
[Proveedor_ManufactureroId] [int] NOT NULL,
[Proveedor_DistribuidorId] [int] NOT NULL,
[Usuario_AprobadorId] [int] NOT NULL,
[Usuario_AdministradorId] [int] NOT NULL,
[Tipo_ContratoId] [int] NOT NULL,
[Método_PagoId] [int] NOT NULL,
[Unidad_NegocioId] [int] NOT NULL,
[Tipo_PagoId] [int] NOT NULL,
[Orden_CompraId] [int] NOT NULL,
[TarjetaId] [int] NOT NULL,
[Area_ContratoId] [int] NOT NULL,
[ProductoId] [int] NOT NULL,
[RecordatorioId] [int] NOT NULL,
[EstadoId] [int] NOT NULL,
[Detalles_Producto] [varchar](max) NULL,
CONSTRAINT [PK_ContratoId] PRIMARY KEY CLUSTERED
(
    [ContratoId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[estados]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[estados](
    [EstadoId] [int] IDENTITY(1,1) NOT NULL,
    [Descripcion] [varchar](25) NOT NULL,
    CONSTRAINT [PK_EstadosId] PRIMARY KEY CLUSTERED
(
    [EstadoId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

```

```
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[metodo_pagos]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[metodo_pagos](
    [Metodo_PagoId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre_Metodo_Pago] [varchar](25) NOT NULL,
    CONSTRAINT [PK_Metodo_PagoId] PRIMARY KEY CLUSTERED
(
    [Metodo_PagoId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[orden_compras]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[orden_compras](
    [Orden_CompraId] [int] IDENTITY(1,1) NOT NULL,
    [Fecha_Ultima_Orden_Compra] [date] NOT NULL,
    [Ultimo_Numero_Factura] [varchar](15) NOT NULL,
    [Codigo_Presupuestario2016] [varchar](15) NOT NULL,
    [Monto_Previo] [varchar](25) NOT NULL,
    [Direccion_Archivo] [varchar](100) NULL DEFAULT (NULL),
    CONSTRAINT [PK_Orden_CompraId] PRIMARY KEY CLUSTERED
(
    [Orden_CompraId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[perfiles]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[perfiles](
    [PerfilId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre_Perfil] [varchar](25) NOT NULL,
    [Descripcion] [varchar](100) NULL DEFAULT (NULL),
    CONSTRAINT [PK_PerfilId] PRIMARY KEY CLUSTERED
```

```

(
    [PerfilId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[permisos]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[permisos](
    [PermisoId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre_Permiso] [varchar](25) NOT NULL,
    [Descripcion] [varchar](100) NULL DEFAULT (NULL),
    CONSTRAINT [PK_PermisoId] PRIMARY KEY CLUSTERED
(
    [PermisoId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[productos]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[productos](
    [ProductoId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre_Producto] [varchar](25) NOT NULL,
    [Precio] [numeric](18, 3) NOT NULL,
    [Descripcion] [varchar](500) NULL DEFAULT (NULL),
    [Direccion_Fotografia] [varchar](100) NULL DEFAULT (NULL),
    [Proveedor_ManufactureroId] [int] NOT NULL,
    [Proveedor_DistribuidorId] [int] NOT NULL,
    CONSTRAINT [PK_ProductoId] PRIMARY KEY CLUSTERED
(
    [ProductoId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[proveedores]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO

```

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[proveedores](
    [ProveedorId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre_Proveedor] [varchar](25) NOT NULL,
    [Correo_Proveedor] [varchar](50) NOT NULL,
    [Nombre_Contacto] [varchar](25) NULL DEFAULT (NULL),
    [Telefono_Contacto] [varchar](20) NULL DEFAULT (NULL),
    [Ubicacion] [varchar](50) NULL DEFAULT (NULL),
    [Descripcion] [varchar](500) NULL DEFAULT (NULL),
    [Tipo_ProveedorId] [int] NOT NULL,
    CONSTRAINT [PK_ProveedorId] PRIMARY KEY CLUSTERED
(
    [ProveedorId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[recordatorios]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[recordatorios](
    [RecordatorioId] [int] IDENTITY(1,1) NOT NULL,
    [Descripcion] [varchar](25) NOT NULL,
    CONSTRAINT [PK_RecordatorioId] PRIMARY KEY CLUSTERED
(
    [RecordatorioId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[tarjetas]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tarjetas](
    [TarjetaId] [int] IDENTITY(1,1) NOT NULL,
    [Ultimos_Digitos_Tarjeta] [varchar](4) NOT NULL,
    [Fecha_Vencimiento_Tarjeta] [date] NOT NULL,
    [Nombre_Persona_Asignada_A_Tarjeta] [varchar](25) NOT NULL,
    [Tipo_TarjetaId] [int] NOT NULL,
    CONSTRAINT [PK_TarjetaId] PRIMARY KEY CLUSTERED
(

```

```

        [TarjetaId] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[tipo_contratos]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tipo_contratos](
    [Tipo_ContratoId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre_Tipo_Contrato] [varchar](25) NOT NULL,
    [Descripcion] [varchar](100) NULL DEFAULT (NULL),
    CONSTRAINT [PK_Tipo_ContratoId] PRIMARY KEY CLUSTERED
(
    [Tipo_ContratoId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[tipo_pagos]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tipo_pagos](
    [Tipo_PagoId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre_Tipo_Pago] [varchar](25) NOT NULL,
    CONSTRAINT [PK_Tipo_PagoId] PRIMARY KEY CLUSTERED
(
    [Tipo_PagoId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[tipo_proveedores]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tipo_proveedores](
    [Tipo_ProveedorId] [int] IDENTITY(1,1) NOT NULL,

```

```

        [Descripcion] [varchar](25) NOT NULL,
    CONSTRAINT [PK_Tipo_ProveedorId] PRIMARY KEY CLUSTERED
    (
        [Tipo_ProveedorId] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[tipo_tarjetas]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tipo_tarjetas](
    [Tipo_TarjetaId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre_Tarjeta] [varchar](25) NOT NULL,
    [Descripcion] [varchar](100) NULL DEFAULT (NULL),
    CONSTRAINT [PK_Tipo_TarjetaId] PRIMARY KEY CLUSTERED
    (
        [Tipo_TarjetaId] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[unidad_negocios]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[unidad_negocios](
    [Unidad_NegocioId] [int] IDENTITY(1,1) NOT NULL,
    [Numero_Unidad_Negocio] [varchar](15) NOT NULL,
    [Nombre_Unidad_Negocio] [varchar](15) NOT NULL,
    [Nombre_Previo_Unidad_Negocio] [varchar](15) NOT NULL,
    CONSTRAINT [PK_Unidad_NegocioId] PRIMARY KEY CLUSTERED
    (
        [Unidad_NegocioId] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[usuarios]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[usuarios](
    [UsuarioId] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](25) NOT NULL,
    [Apellido] [varchar](25) NOT NULL,
    [email] [varchar](50) NOT NULL,
    [pass] [varchar](250) NOT NULL,
    [Telefono] [varchar](20) NULL,
    [Codigo_Usuario] [varchar](15) NOT NULL,
    [username] [varchar](30) NOT NULL,
    [Fecha_Creacion] [date] NOT NULL,
    [Direccion_Fotografia] [varchar](100) NULL DEFAULT (NULL),
    [PerfilId] [int] NOT NULL,
    [pass2] [varbinary](max) NULL,
    CONSTRAINT [PK_UsuarioId] PRIMARY KEY CLUSTERED
(
    [UsuarioId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[usuarios_perfiles_permisos]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[usuarios_perfiles_permisos](
    [Usuario_Perfil_PermisoId] [int] IDENTITY(1,1) NOT NULL,
    [UsuarioId] [int] NOT NULL,
    [PerfilId] [int] NOT NULL,
    [PermisoId] [int] NOT NULL,
    [Descripcion] [varchar](100) NULL DEFAULT (NULL),
    CONSTRAINT [PK_Usuario_Perfil_PermisosId] PRIMARY KEY CLUSTERED
(
    [Usuario_Perfil_PermisoId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Nombre_Contrato2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Numero_Contrato2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Numero_Servicio_Contrato2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Costo_Total2]
GO

```



```

ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Costo_Por_Cada_Pago2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Cantidad_Licencias2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Garantia2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Fecha_Creacion2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Descripcion2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Direccion_Archivo2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Proveedor_ManufactureroId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Proveedor_DistribuidorId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Usuario_AprobadorId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Usuario_AdministradorId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Tipo_ContratoId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Metodo_PagoId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Unidad_NegocioId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Tipo_PagoId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Orden_CompraId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [TarjetaId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Area_ContratoId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [ProductoId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [RecordatorioId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [EstadoId2]
GO
ALTER TABLE [dbo].[Borradorcontratos] ADD DEFAULT (NULL) FOR [Detalles_Producto2]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT
[FK2_Area_Contratos_PK_Area_Contrato] FOREIGN KEY([Area_ContratoId2])
REFERENCES [dbo].[area_contratos] ([Area_ContratoId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Area_Contratos_PK_Area_Contrato]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT [FK2_Estados_EstadoId] FOREIGN
KEY([EstadoId2])
REFERENCES [dbo].[estados] ([EstadoId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Estados_EstadoId]
GO

```

```

ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT
[FK2_Metodo_Pagos_PK_Metodo_PagoId] FOREIGN KEY([Metodo_PagoId2])
REFERENCES [dbo].[metodo_pagos] ([Metodo_PagoId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Metodo_Pagos_PK_Metodo_PagoId]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT
[FK2_Orden_Compras_PK_Orden_CompraId] FOREIGN KEY([Orden_CompraId2])
REFERENCES [dbo].[orden_compras] ([Orden_CompraId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Orden_Compras_PK_Orden_CompraId]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT [FK2_Productos_ProductoId] FOREIGN
KEY([ProductoId2])
REFERENCES [dbo].[productos] ([ProductoId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Productos_ProductoId]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT
[FK2_Proveedores_Distribuidor_PK_ProveedorId] FOREIGN KEY([Proveedor_DistribuidorId2])
REFERENCES [dbo].[proveedores] ([ProveedorId])
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Proveedores_Distribuidor_PK_ProveedorId]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT
[FK2_Proveedores_Manufacturero_PK_ProveedorId] FOREIGN KEY([Proveedor_ManufactureroId2])
REFERENCES [dbo].[proveedores] ([ProveedorId])
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT
[FK2_Proveedores_Manufacturero_PK_ProveedorId]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT [FK2_Recordatorios_RecordatorioId]
FOREIGN KEY([RecordatorioId2])
REFERENCES [dbo].[recordatorios] ([RecordatorioId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Recordatorios_RecordatorioId]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT [FK2_Tarjetas_Pk_TarjetaId]
FOREIGN KEY([TarjetaId2])
REFERENCES [dbo].[tarjetas] ([TarjetaId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Tarjetas_Pk_TarjetaId]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT
[FK2_Tipo_Contratos_PK_Tipo_ContratoId] FOREIGN KEY([Tipo_ContratoId2])
REFERENCES [dbo].[tipo_contratos] ([Tipo_ContratoId])
ON UPDATE CASCADE
ON DELETE CASCADE

```

```

GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Tipo_Contratos_PK_Tipo_ContratoId]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT [FK2_Tipo_Pagos_PK_Tipo_PagoId]
FOREIGN KEY([Tipo_PagoId2])
REFERENCES [dbo].[tipo_pagos] ([Tipo_PagoId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Tipo_Pagos_PK_Tipo_PagoId]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT
[FK2_Unidad_Negocios_PK_Unidad_NegocioId] FOREIGN KEY([Unidad_NegocioId2])
REFERENCES [dbo].[unidad_negocios] ([Unidad_NegocioId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Unidad_Negocios_PK_Unidad_NegocioId]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT
[FK2_Usuarios_Administrador_UsuarioId] FOREIGN KEY([Usuario_AdministradorId2])
REFERENCES [dbo].[usuarios] ([UsuarioId])
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Usuarios_Administrador_UsuarioId]
GO
ALTER TABLE [dbo].[Borradorcontratos] WITH CHECK ADD CONSTRAINT
[FK2_Usuarios_Aprobador_PK_UsuarioId] FOREIGN KEY([Usuario_AprobadorId2])
REFERENCES [dbo].[usuarios] ([UsuarioId])
GO
ALTER TABLE [dbo].[Borradorcontratos] CHECK CONSTRAINT [FK2_Usuarios_Aprobador_PK_UsuarioId]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT [FK_Area_Contratos_PK_Area_Contrato]
FOREIGN KEY([Area_ContratoId])
REFERENCES [dbo].[area_contratos] ([Area_ContratoId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Area_Contratos_PK_Area_Contrato]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT [FK_Estados_EstadoId] FOREIGN
KEY([EstadoId])
REFERENCES [dbo].[estados] ([EstadoId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Estados_EstadoId]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT [FK_Metodo_Pagos_PK_Metodo_PagoId] FOREIGN
KEY([Metodo_PagoId])
REFERENCES [dbo].[metodo_pagos] ([Metodo_PagoId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Metodo_Pagos_PK_Metodo_PagoId]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT [FK_Orden_Compras_PK_Orden_CompraId]
FOREIGN KEY([Orden_CompraId])
REFERENCES [dbo].[orden_compras] ([Orden_CompraId])

```

```
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Orden_Compras_PK_Orden_CompraId]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT [FK_Productos_ProductoId] FOREIGN
KEY([ProductoId])
REFERENCES [dbo].[productos] ([ProductoId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Productos_ProductoId]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT
[FK_Proveedores_Distribuidor_PK_ProveedorId] FOREIGN KEY([Proveedor_DistribuidorId])
REFERENCES [dbo].[proveedores] ([ProveedorId])
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Proveedores_Distribuidor_PK_ProveedorId]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT
[FK_Proveedores_Manufacturero_PK_ProveedorId] FOREIGN KEY([Proveedor_ManufactureroId])
REFERENCES [dbo].[proveedores] ([ProveedorId])
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Proveedores_Manufacturero_PK_ProveedorId]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT [FK_Recordatorios_RecordatorioId] FOREIGN
KEY([RecordatorioId])
REFERENCES [dbo].[recordatorios] ([RecordatorioId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Recordatorios_RecordatorioId]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT [FK_Tarjetas_Pk_TarjetaId] FOREIGN
KEY([TarjetaId])
REFERENCES [dbo].[tarjetas] ([TarjetaId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Tarjetas_Pk_TarjetaId]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT [FK_Tipo_Contratos_PK_Tipo_ContratoId]
FOREIGN KEY([Tipo_ContratoId])
REFERENCES [dbo].[tipo_contratos] ([Tipo_ContratoId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Tipo_Contratos_PK_Tipo_ContratoId]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT [FK_Tipo_Pagos_PK_Tipo_PagoId] FOREIGN
KEY([Tipo_PagoId])
REFERENCES [dbo].[tipo_pagos] ([Tipo_PagoId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Tipo_Pagos_PK_Tipo_PagoId]
GO
```

```
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT [FK_Unidad_Negocios_PK_Unidad_NegocioId]
FOREIGN KEY([Unidad_NegocioId])
REFERENCES [dbo].[unidad_negocios] ([Unidad_NegocioId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Unidad_Negocios_PK_Unidad_NegocioId]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT [FK_Usuarios_Administrador_UsuarioId]
FOREIGN KEY([Usuario_AdministradorId])
REFERENCES [dbo].[usuarios] ([UsuarioId])
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Usuarios_Administrador_UsuarioId]
GO
ALTER TABLE [dbo].[contratos] WITH CHECK ADD CONSTRAINT [FK_Usuarios_Aprobador_PK_UsuarioId]
FOREIGN KEY([Usuario_AprobadorId])
REFERENCES [dbo].[usuarios] ([UsuarioId])
GO
ALTER TABLE [dbo].[contratos] CHECK CONSTRAINT [FK_Usuarios_Aprobador_PK_UsuarioId]
GO
ALTER TABLE [dbo].[productos] WITH CHECK ADD CONSTRAINT
[FK_Proveedor_Distribuidores_PK_ProductoId] FOREIGN KEY([Proveedor_DistribuidorId])
REFERENCES [dbo].[proveedores] ([ProveedorId])
GO
ALTER TABLE [dbo].[productos] CHECK CONSTRAINT [FK_Proveedor_Distribuidores_PK_ProductoId]
GO
ALTER TABLE [dbo].[productos] WITH CHECK ADD CONSTRAINT
[FK_Proveedor_Manufactureros_PK_ProductoId] FOREIGN KEY([Proveedor_ManufactureroId])
REFERENCES [dbo].[proveedores] ([ProveedorId])
GO
ALTER TABLE [dbo].[productos] CHECK CONSTRAINT [FK_Proveedor_Manufactureros_PK_ProductoId]
GO
ALTER TABLE [dbo].[proveedores] WITH CHECK ADD CONSTRAINT [FK_Tipo_Proveedores_PK_ProveedorId]
FOREIGN KEY([Tipo_ProveedorId])
REFERENCES [dbo].[tipo_proveedores] ([Tipo_ProveedorId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[proveedores] CHECK CONSTRAINT [FK_Tipo_Proveedores_PK_ProveedorId]
GO
ALTER TABLE [dbo].[tarjetas] WITH CHECK ADD CONSTRAINT [FK_Tipo_Tarjetas_Tipo_TarjetaId] FOREIGN
KEY([Tipo_TarjetaId])
REFERENCES [dbo].[tipo_tarjetas] ([Tipo_TarjetaId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[tarjetas] CHECK CONSTRAINT [FK_Tipo_Tarjetas_Tipo_TarjetaId]
GO
ALTER TABLE [dbo].[usuarios] WITH CHECK ADD CONSTRAINT [FK_Perfiles_PK_Perfil] FOREIGN
KEY([PerfilId])
REFERENCES [dbo].[perfiles] ([PerfilId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[usuarios] CHECK CONSTRAINT [FK_Perfiles_PK_Perfil]
GO
ALTER TABLE [dbo].[usuarios_perfiles_permisos] WITH CHECK ADD CONSTRAINT [FK_Perfiles_PK_PerfilId]
FOREIGN KEY([PerfilId])
```

```

REFERENCES [dbo].[perfiles] ([PerfilId])
GO
ALTER TABLE [dbo].[usuarios_perfiles_permisos] CHECK CONSTRAINT [FK_Perfiles_PK_PerfilId]
GO
ALTER TABLE [dbo].[usuarios_perfiles_permisos] WITH CHECK ADD CONSTRAINT
[FK_Permisos_PK_PermisoId] FOREIGN KEY([PermisoId])
REFERENCES [dbo].[permisos] ([PermisoId])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[usuarios_perfiles_permisos] CHECK CONSTRAINT [FK_Permisos_PK_PermisoId]
GO
ALTER TABLE [dbo].[usuarios_perfiles_permisos] WITH CHECK ADD CONSTRAINT
[FK_Usuarios_PK_UsuarioId] FOREIGN KEY([UsuarioId])
REFERENCES [dbo].[usuarios] ([UsuarioId])
GO
ALTER TABLE [dbo].[usuarios_perfiles_permisos] CHECK CONSTRAINT [FK_Usuarios_PK_UsuarioId]
GO
/***** Object: StoredProcedure [dbo].[AboutToFinishContractByAreaForReportPDF]    Script Date:
7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[AboutToFinishContractByAreaForReportPDF]
@CodigoArea int
AS
Begin

SELECT contratos.Nombre_Contrato, contratos.Numero_Contrato, contratos.Numero_Servicio_Contrato,
contratos.Fecha_Inicio, contratos.Fecha_Finalizacion,

tipo_pagos.Nombre_Tipo_Pago,contratos.Costo_Total,contratos.Costo_Por_Cada_Pago ,contratos.Fecha_Ren
ovacion_Pago, contratos.Cantidad_Licencias, contratos.Garantia,
contratos.Fecha_Vencimiento_Garantia,
unidad_negocios.Nombre_Unidad_Negocio, metodo_pagos.Nombre_Metodo_Pago,

tarjetas.Nombre_Persona_Asignada_A_Tarjeta,tarjetas.Ultimos_Digitos_Tarjeta,tarjetas.Fecha_Vencimien
to_Tarjeta, productos.Nombre_Producto,
tipo_contratos.Nombre_Tipo_Contrato, Detalles_Producto
FROM
contratos INNER JOIN
estados ON contratos.EstadoId = estados.EstadoId INNER JOIN
metodo_pagos ON contratos.Metodo_PagoId = metodo_pagos.Metodo_PagoId INNER JOIN
orden_compras ON contratos.Orden_CompraId = orden_compras.Orden_CompraId INNER JOIN
productos ON contratos.ProductoId = productos.ProductoId INNER JOIN
tarjetas ON contratos.TarjetaId = tarjetas.TarjetaId INNER JOIN
tipo_contratos ON contratos.Tipo_ContratoId = tipo_contratos.Tipo_ContratoId INNER JOIN
tipo_pagos ON contratos.Tipo_PagoId = tipo_pagos.Tipo_PagoId INNER JOIN
unidad_negocios ON contratos.Unidad_NegocioId = unidad_negocios.Unidad_NegocioId

where dbo.estados.Descripcion= 'Proximo a Vencer' and dbo.contratos.Area_ContratoId = @CodigoArea;
end;
GO
/***** Object: StoredProcedure [dbo].[AboutToFinishContractForReportPDF]    Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
CREATE procedure [dbo].[AboutToFinishContractForReportPDF]
AS
Begin

SELECT  contratos.Nombre_Contrato, contratos.Numero_Contrato, contratos.Numero_Servicio_Contrato,
contratos.Fecha_Inicio, contratos.Fecha_Finalizacion,

tipo_pagos.Nombre_Tipo_Pago,contratos.Costo_Total,contratos.Costo_Por_Cada_Pago ,contratos.Fecha_Ren
ovacion_Pago, contratos.Cantidad_Licencias, contratos.Garantia,
contratos.Fecha_Vencimiento_Garantia,
        unidad_negocios.Nombre_Unidad_Negocio, metodo_pagos.Nombre_Metodo_Pago,

tarjetas.Nombre_Persona_Asignada_A_Tarjeta,tarjetas.Ultimos_Digitos_Tarjeta,tarjetas.Fecha_Vencimien
to_Tarjeta, productos.Nombre_Producto,
        tipo_contratos.Nombre_Tipo_Contrato, Detalles_Producto
FROM      contratos INNER JOIN
        estados ON contratos.EstadoId = estados.EstadoId INNER JOIN
        metodo_pagos ON contratos.Metodo_PagoId = metodo_pagos.Metodo_PagoId INNER JOIN
        orden_compras ON contratos.Orden_CompraId = orden_compras.Orden_CompraId INNER JOIN
        productos ON contratos.ProductoId = productos.ProductoId INNER JOIN
        tarjetas ON contratos.TarjetaId = tarjetas.TarjetaId INNER JOIN
        tipo_contratos ON contratos.Tipo_ContratoId = tipo_contratos.Tipo_ContratoId INNER JOIN
        tipo_pagos ON contratos.Tipo_PagoId = tipo_pagos.Tipo_PagoId INNER JOIN
        unidad_negocios ON contratos.Unidad_NegocioId = unidad_negocios.Unidad_NegocioId

        where dbo.estados.Descripcion= 'Proximo a vencer';

end;
GO
/***** Object:  StoredProcedure [dbo].[ActiveContractByAreaForReportPDF]      Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[ActiveContractByAreaForReportPDF]
@CodigoArea int
AS
Begin

SELECT  contratos.Nombre_Contrato, contratos.Numero_Contrato, contratos.Numero_Servicio_Contrato,
contratos.Fecha_Inicio, contratos.Fecha_Finalizacion,

tipo_pagos.Nombre_Tipo_Pago,contratos.Costo_Total,contratos.Costo_Por_Cada_Pago ,contratos.Fecha_Ren
ovacion_Pago, contratos.Cantidad_Licencias, contratos.Garantia,
contratos.Fecha_Vencimiento_Garantia,
        unidad_negocios.Nombre_Unidad_Negocio, metodo_pagos.Nombre_Metodo_Pago,

tarjetas.Nombre_Persona_Asignada_A_Tarjeta,tarjetas.Ultimos_Digitos_Tarjeta,tarjetas.Fecha_Vencimien
to_Tarjeta, productos.Nombre_Producto,
        tipo_contratos.Nombre_Tipo_Contrato, Detalles_Producto
FROM      contratos INNER JOIN
        estados ON contratos.EstadoId = estados.EstadoId INNER JOIN
        metodo_pagos ON contratos.Metodo_PagoId = metodo_pagos.Metodo_PagoId INNER JOIN
        orden_compras ON contratos.Orden_CompraId = orden_compras.Orden_CompraId INNER JOIN
        productos ON contratos.ProductoId = productos.ProductoId INNER JOIN

```

```

tarjetas ON contratos.TarjetaId = tarjetas.TarjetaId INNER JOIN
tipo_contratos ON contratos.Tipo_ContratoId = tipo_contratos.Tipo_ContratoId INNER JOIN
tipo_pagos ON contratos.Tipo_PagoId = tipo_pagos.Tipo_PagoId INNER JOIN
unidad_negocios ON contratos.Unidad_NegocioId = unidad_negocios.Unidad_NegocioId

where dbo.estados.Descripcion= 'Activo' and dbo.contratos.Area_ContratoId =
@CodigoArea;

end;
GO
/***** Object: StoredProcedure [dbo].[ActiveContractForReportPDF]    Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[ActiveContractForReportPDF]
AS
Begin

SELECT  contratos.Nombre_Contrato, contratos.Numero_Contrato, contratos.Numero_Servicio_Contrato,
contratos.Fecha_Inicio, contratos.Fecha_Finalizacion,

tipo_pagos.Nombre_Tipo_Pago, contratos.Costo_Total, contratos.Costo_Por_Cada_Pago , contratos.Fecha_Ren
ovacion_Pago, contratos.Cantidad_Licencias, contratos.Garantia,
contratos.Fecha_Vencimiento_Garantia,
unidad_negocios.Nombre_Unidad_Negocio, metodo_pagos.Nombre_Metodo_Pago,

tarjetas.Nombre_Persona_Asignada_A_Tarjeta, tarjetas.Ultimos_Digitos_Tarjeta, tarjetas.Fecha_Vencimien
to_Tarjeta, productos.Nombre_Producto,
tipo_contratos.Nombre_Tipo_Contrato, Detalles_Producto
FROM    contratos INNER JOIN
estados ON contratos.EstadoId = estados.EstadoId INNER JOIN
metodo_pagos ON contratos.Metodo_PagoId = metodo_pagos.Metodo_PagoId INNER JOIN
orden_compras ON contratos.Orden_CompraId = orden_compras.Orden_CompraId INNER JOIN
productos ON contratos.ProductoId = productos.ProductoId INNER JOIN
tarjetas ON contratos.TarjetaId = tarjetas.TarjetaId INNER JOIN
tipo_contratos ON contratos.Tipo_ContratoId = tipo_contratos.Tipo_ContratoId INNER JOIN
tipo_pagos ON contratos.Tipo_PagoId = tipo_pagos.Tipo_PagoId INNER JOIN
unidad_negocios ON contratos.Unidad_NegocioId = unidad_negocios.Unidad_NegocioId

where dbo.estados.Descripcion= 'Activo';

end;
GO
/***** Object: StoredProcedure [dbo].[AllDetailsContractByAreaForReportPDF]    Script Date:
7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE procedure [dbo].[AllDetailsContractByAreaForReportPDF]
@CodigoArea int
AS
Begin

```



```

SELECT  contratos.Nombre_Contrato, contratos.Numero_Contrato, contratos.Numero_Servicio_Contrato,
contratos.Fecha_Inicio, contratos.Fecha_Finalizacion,

tipo_pagos.Nombre_Tipo_Pago,contratos.Costo_Total,contratos.Costo_Por_Cada_Pago ,contratos.Fecha_Ren
ovacion_Pago, contratos.Cantidad_Licencias, contratos.Garantia,
contratos.Fecha_Vencimiento_Garantia,
        unidad_negocios.Nombre_Unidad_Negocio, estados.Descripcion as Estado_Contrato,
metodo_pagos.Nombre_Metodo_Pago,

tarjetas.Nombre_Persona_Asignada_A_Tarjeta,tarjetas.Ultimos_Digitos_Tarjeta,tarjetas.Fecha_Vencimien
to_Tarjeta, productos.Nombre_Producto,
        tipo_contratos.Nombre_Tipo_Contrato, Detalles_Producto
FROM      contratos INNER JOIN
        estados ON contratos.EstadoId = estados.EstadoId INNER JOIN
        metodo_pagos ON contratos.Metodo_PagoId = metodo_pagos.Metodo_PagoId INNER JOIN
        orden_compras ON contratos.Orden_CompraId = orden_compras.Orden_CompraId INNER JOIN
        productos ON contratos.ProductoId = productos.ProductoId INNER JOIN
        tarjetas ON contratos.TarjetaId = tarjetas.TarjetaId INNER JOIN
        tipo_contratos ON contratos.Tipo_ContratoId = tipo_contratos.Tipo_ContratoId INNER JOIN
        tipo_pagos ON contratos.Tipo_PagoId = tipo_pagos.Tipo_PagoId INNER JOIN
        unidad_negocios ON contratos.Unidad_NegocioId = unidad_negocios.Unidad_NegocioId

        where dbo.contratos.Area_ContratoId = @CodigoArea;

end;
GO
/***** Object:  StoredProcedure [dbo].[AllDetailsContractForReportPDF]      Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE procedure [dbo].[AllDetailsContractForReportPDF]
AS
Begin

```

```

SELECT  contratos.Nombre_Contrato, contratos.Numero_Contrato, contratos.Numero_Servicio_Contrato,
contratos.Fecha_Inicio, contratos.Fecha_Finalizacion,

tipo_pagos.Nombre_Tipo_Pago,contratos.Costo_Total,contratos.Costo_Por_Cada_Pago ,contratos.Fecha_Ren
ovacion_Pago, contratos.Cantidad_Licencias, contratos.Garantia,
contratos.Fecha_Vencimiento_Garantia,
        unidad_negocios.Nombre_Unidad_Negocio, estados.Descripcion as Estado_Contrato,
metodo_pagos.Nombre_Metodo_Pago,

tarjetas.Nombre_Persona_Asignada_A_Tarjeta,tarjetas.Ultimos_Digitos_Tarjeta,tarjetas.Fecha_Vencimien
to_Tarjeta, productos.Nombre_Producto,
        tipo_contratos.Nombre_Tipo_Contrato, Detalles_Producto
FROM      contratos INNER JOIN
        estados ON contratos.EstadoId = estados.EstadoId INNER JOIN
        metodo_pagos ON contratos.Metodo_PagoId = metodo_pagos.Metodo_PagoId INNER JOIN
        orden_compras ON contratos.Orden_CompraId = orden_compras.Orden_CompraId INNER JOIN
        productos ON contratos.ProductoId = productos.ProductoId INNER JOIN
        tarjetas ON contratos.TarjetaId = tarjetas.TarjetaId INNER JOIN
        tipo_contratos ON contratos.Tipo_ContratoId = tipo_contratos.Tipo_ContratoId INNER JOIN

```

```

    tipo_pagos ON contratos.Tipo_PagoId = tipo_pagos.Tipo_PagoId INNER JOIN
    unidad_negocios ON contratos.Unidad_NegocioId = unidad_negocios.Unidad_NegocioId

end;
GO
/***** Object: StoredProcedure [dbo].[defeatedContractByAreaForReportPDF]    Script Date:
7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE procedure [dbo].[defeatedContractByAreaForReportPDF]
@CodigoArea int
AS
Begin

SELECT  contratos.Nombre_Contrato, contratos.Numero_Contrato, contratos.Numero_Servicio_Contrato,
contratos.Fecha_Inicio, contratos.Fecha_Finalizacion,

tipo_pagos.Nombre_Tipo_Pago,contratos.Costo_Total,contratos.Costo_Por_Cada_Pago ,contratos.Fecha_Ren
ovacion_Pago, contratos.Cantidad_Licencias, contratos.Garantia,
contratos.Fecha_Vencimiento_Garantia,
        unidad_negocios.Nombre_Unidad_Negocio, metodo_pagos.Nombre_Metodo_Pago,

tarjetas.Nombre_Persona_Asignada_A_Tarjeta,tarjetas.Ultimos_Digitos_Tarjeta,tarjetas.Fecha_Vencimien
to_Tarjeta, productos.Nombre_Producto,
        tipo_contratos.Nombre_Tipo_Contrato, Detalles_Producto
FROM      contratos INNER JOIN
        estados ON contratos.EstadoId = estados.EstadoId INNER JOIN
        metodo_pagos ON contratos.Metodo_PagoId = metodo_pagos.Metodo_PagoId INNER JOIN
        orden_compras ON contratos.Orden_CompraId = orden_compras.Orden_CompraId INNER JOIN
        productos ON contratos.ProductoId = productos.ProductoId INNER JOIN
        tarjetas ON contratos.TarjetaId = tarjetas.TarjetaId INNER JOIN
        tipo_contratos ON contratos.Tipo_ContratoId = tipo_contratos.Tipo_ContratoId INNER JOIN
        tipo_pagos ON contratos.Tipo_PagoId = tipo_pagos.Tipo_PagoId INNER JOIN
        unidad_negocios ON contratos.Unidad_NegocioId = unidad_negocios.Unidad_NegocioId

        where dbo.estados.Descripcion= 'Vencido' and dbo.contratos.Area_ContratoId =
@CodigoArea;

end;
GO
/***** Object: StoredProcedure [dbo].[defeatedContractForReportPDF]    Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE procedure [dbo].[defeatedContractForReportPDF]
AS
Begin

SELECT  contratos.Nombre_Contrato, contratos.Numero_Contrato, contratos.Numero_Servicio_Contrato,
contratos.Fecha_Inicio, contratos.Fecha_Finalizacion,

```

```

tipo_pagos.Nombre_Tipo_Pago,contratos.Costo_Total,contratos.Costo_Por_Cada_Pago ,contratos.Fecha_Ren
ovacion_Pago, contratos.Cantidad_Licencias, contratos.Garantia,
contratos.Fecha_Vencimiento_Garantia,
        unidad_negocios.Nombre_Unidad_Negocio, metodo_pagos.Nombre_Metodo_Pago,

tarjetas.Nombre_Persona_Asignada_A_Tarjeta,tarjetas.Ultimos_Digitos_Tarjeta,tarjetas.Fecha_Vencimien
to_Tarjeta, productos.Nombre_Producto,
        tipo_contratos.Nombre_Tipo_Contrato, Detalles_Producto
FROM      contratos INNER JOIN
        estados ON contratos.EstadoId = estados.EstadoId INNER JOIN
        metodo_pagos ON contratos.Metodo_PagoId = metodo_pagos.Metodo_PagoId INNER JOIN
        orden_compras ON contratos.Orden_CompraId = orden_compras.Orden_CompraId INNER JOIN
        productos ON contratos.ProductoId = productos.ProductoId INNER JOIN
        tarjetas ON contratos.TarjetaId = tarjetas.TarjetaId INNER JOIN
        tipo_contratos ON contratos.Tipo_ContratoId = tipo_contratos.Tipo_ContratoId INNER JOIN
        tipo_pagos ON contratos.Tipo_PagoId = tipo_pagos.Tipo_PagoId INNER JOIN
        unidad_negocios ON contratos.Unidad_NegocioId = unidad_negocios.Unidad_NegocioId

        where dbo.estados.Descripcion= 'Vencido';

end;
GO
/***** Object:  StoredProcedure [dbo].[SP_DeleteAreaContratos]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_DeleteAreaContratos]
        @AreaContratosId int
    AS
    Begin
        delete from area_contratos
        where Area_ContratoId= @AreaContratosId;
    END;

GO
/***** Object:  StoredProcedure [dbo].[SP_DeleteContract]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_DeleteContract]
        @ContratoId int
    AS
    Begin
        Delete from contratos where ContratoId=@ContratoId;
    End;

GO
/***** Object:  StoredProcedure [dbo].[SP_DeleteDraftContract]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

create procedure [dbo].[SP_DeleteDraftContract]
@ContratoId int
AS
Begin
    Delete from Borradorcontratos where ContratoId2=@ContratoId;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_DeleteMethodPaymentById]    Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_DeleteMethodPaymentById]
@MetodoPagoId int
AS
Begin
    Delete FROM metodo_pagos
        where Metodo_PagoId= @MetodoPagoId;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_DeleteOrdenCompras]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_DeleteOrdenCompras]
@OrdenCompraId int
AS
Begin
    Delete from orden_compras where Orden_CompraId=@OrdenCompraId
END;

/*Fin SP para tabla Orden Compras */

GO
/***** Object: StoredProcedure [dbo].[SP_DeleteProduct]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_DeleteProduct]
@ProductId int
AS
Begin
    delete from productos where ProductoId= @ProductId;
END;

GO
/***** Object: StoredProcedure [dbo].[SP_DeleteProvider]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO

```

```
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_DeleteProvider]
    @ProveedorId int
AS
Begin
    delete from proveedores where ProveedorId= @ProveedorId;
END;

GO
/***** Object:  StoredProcedure [dbo].[SP_DeleteTarjetas]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_DeleteTarjetas]
    @TarjetaId int
AS
Begin
    delete from tarjetas
    where TarjetaId= @TarjetaId;
END;

/*Fin SP para tabla Tarjetas */

GO
/***** Object:  StoredProcedure [dbo].[SP_DeleteTipoContratos]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_DeleteTipoContratos]
    @TipoContratosId int
AS
Begin
    delete from tipo_contratos
    where Tipo_ContratoId= @TipoContratosId;
END;

/*Fin SP para tabla Tipo Contratos */

GO
/***** Object:  StoredProcedure [dbo].[SP_DeleteUnidadNegocios]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_DeleteUnidadNegocios]
    @UnidadNegocioId int
AS
Begin
    delete from unidad_negocios
    where Unidad_NegocioId=@UnidadNegocioId;
```

```
END;

/*Fin SP para tabla Unidad Negocios */
GO
/***** Object: StoredProcedure [dbo].[SP_DeleteUser]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_DeleteUser]
    @UsuarioId int
AS
Begin

    Delete from usuarios_perfiles_permisos where usuarios_perfiles_permisos.UsuarioId =@UsuarioId;
    Delete from usuarios where UsuarioId= @UsuarioId;

END;

GO
/***** Object: StoredProcedure [dbo].[SP_EditAreaContratos]    Script Date: 7/20/2016 2:01:57 PM *****/
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_EditAreaContratos]
    @AreaContratosId int,
    @Nombre_Area varchar(25)
AS
Begin

UPDATE area_contratos
    set Nombre_Area=@Nombre_Area
    where Area_ContratoId= @AreaContratosId;
END;

GO
/***** Object: StoredProcedure [dbo].[SP_EditContratos]    Script Date: 7/20/2016 2:01:57 PM *****/
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_EditContratos]

    @ContratoId int,
    @Nombre_Contrato varchar(25),
    @Numero_Contrato varchar(15),
    @Numero_Servicio_Contrato varchar(15),
    @Fecha_Inicio Date,
    @Fecha_Finalizacion Date,
    @Costo_Total numeric(18,2),
    @Costo_Por_Cada_Pago numeric (18,2),
    @Fecha_Renovacion_Pago Date,
    @Cantidad_Licencias int,
    @Garantia varchar(25),
    @Fecha_Vencimiento_Garantia Date,
```

```

@Fecha_Creacion Date,
@Descripcion varchar(100),
@Direccion_Archivo varchar(100),
@Proveedor_ManufactureroId int,
@Proveedor_Distribuidor int,
@Usuario_AprobadorId int,
@Usuario_AdministradorId int,
@Tipo_ContratoId int,
@Metodo_PagoId int,
@Unidad_NegocioId int,
@Tipo_PagoId int,
@Orden_CompraId int,
@TarjetaId int,
@Area_ContratoId int,
@ProductoId int,
@RecordatorioId int,
@EstadoId int,
@Detalles_Producto varchar(max)

```

AS

Begin

UPDATE contratos

```

set Nombre_Contrato=@Nombre_Contrato, Numero_Contrato=@Numero_Contrato,
Numero_Servicio_Contrato=@Numero_Servicio_Contrato, Fecha_Inicio=@Fecha_Inicio, Fecha_Finalizacion=@Fe
cha_Finalizacion,

```

```

Costo_Total=@Costo_Total, Costo_Por_Cada_Pago=@Costo_Por_Cada_Pago, Fecha_Renovacion_Pago=@Fecha_Renov
acion_Pago, Cantidad_Licencias=@Cantidad_Licencias, Garantia=@Garantia,

```

```

Fecha_Vencimiento_Garantia=@Fecha_Vencimiento_Garantia, Fecha_Creacion=@Fecha_Creacion, Descripcion=@D
escripcion, Direccion_Archivo=@Direccion_Archivo, Proveedor_ManufactureroId=@Proveedor_ManufactureroId
,

```

```

Proveedor_DistribuidorId=@Proveedor_Distribuidor, Usuario_AprobadorId=@Usuario_AprobadorId, Usuario_Ad
ministradorId=@Usuario_AdministradorId, Tipo_ContratoId=@Tipo_ContratoId, Metodo_PagoId=@Metodo_PagoId
,

```

```

Unidad_NegocioId=@Unidad_NegocioId, Tipo_PagoId=@Tipo_PagoId, Orden_CompraId=@Orden_CompraId, TarjetaId
=@TarjetaId, Area_ContratoId=@Area_ContratoId, ProductoId=@ProductoId, RecordatorioId=@RecordatorioId, E
stadoId=@EstadoId,
Detalles_Producto=@Detalles_Producto

```

```

where ContratoId= @ContratoId;

```

END;

GO

/\*\*\*\*\* Object: StoredProcedure [dbo].[SP\_EditMethodPaymentById] Script Date: 7/20/2016 2:01:57

PM \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```

create procedure [dbo].[SP_EditMethodPaymentById]
@MetodoPagoId int,
@Nombre_Metodo_Pago varchar(25)

```

AS

```

Begin
    UPDATE metodo_pagos
    set Nombre_Metodo_Pago=@Nombre_Metodo_Pago
    where Metodo_PagoId= @MetodoPagoId;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_EditOrdenCompras]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_EditOrdenCompras]
@OrdenCompraId int,
@Fecha_Ultima_Orden_Compra Date,
@Ultimo_Numero_Factura varchar(15),
@Codigo_Presupuestario_2016 varchar(15),
@Monto_Previo varchar(25)

AS
Begin
    UPDATE orden_compras
    set Fecha_Ultima_Orden_Compra=@Fecha_Ultima_Orden_Compra,
    Ultimo_Numero_Factura=@Ultimo_Numero_Factura,Codigo_Presupuestario2016=@Codigo_Presupuestario_2016,
    Monto_Previo=@Monto_Previo
    where Orden_CompraId= @OrdenCompraId;
END;

/*Fin SP para tabla Orden Compras */

/* SP para tabla Perfiles */

GO
/***** Object:  StoredProcedure [dbo].[SP_EditProduct]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_EditProduct]

@ProductoId int,
@Nombre_Producto varchar(25),
@Precio numeric(18,2),
@Descripcion varchar(100),
@Proveedor_ManufactureroId int,
@Proveedor_DistribuidorId int

AS
Begin
    UPDATE productos
    set Nombre_Producto=@Nombre_Producto, Precio=@Precio,
    Descripcion=@Descripcion,Proveedor_ManufactureroId=@Proveedor_ManufactureroId,
    Proveedor_DistribuidorId=@Proveedor_DistribuidorId
    where ProductoId=@ProductoId;

```



END;

GO  
 /\*\*\*\*\* Object: StoredProcedure [dbo].[SP\_EditProvider]      Script Date: 7/20/2016 2:01:57 PM  
 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE procedure [dbo].[SP\_EditProvider]

@ProveedorId int,  
 @Nombre\_Proveedor varchar(25),  
 @Correo\_Proveedor varchar(50),  
 @Nombre\_Contacto varchar(25),  
 @Telefono\_Contacto varchar(20),  
 @Ubicacion varchar(50),  
 @Descripcion varchar(100),  
 @Tipo\_ProveedorId int

AS

Begin

UPDATE proveedores

set Nombre\_Proveedor=@Nombre\_Proveedor, Correo\_Proveedor=@Correo\_Proveedor,  
 Nombre\_Contacto=@Nombre\_Contacto, Telefono\_Contacto=@Telefono\_Contacto,  
 Ubicacion=@Ubicacion, Descripcion=@Descripcion, Tipo\_ProveedorId=@Tipo\_ProveedorId  
 where ProveedorId= @ProveedorId;

END;

GO  
 /\*\*\*\*\* Object: StoredProcedure [dbo].[SP\_EditTarjetas]      Script Date: 7/20/2016 2:01:57 PM  
 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create procedure [dbo].[SP\_EditTarjetas]  
 @TarjetaId int,  
 @Ultimos\_Digitos\_Tarjeta varchar(4),  
 @Fecha\_Vencimiento\_Tarjeta Date,  
 @Nombre\_Persona\_Asignada\_A\_Yarjeta varchar(25),  
 @Tipo\_TarjetaId int

AS

Begin

UPDATE tarjetas

set Ultimos\_Digitos\_Tarjeta=@Ultimos\_Digitos\_Tarjeta,  
 Fecha\_Vencimiento\_Tarjeta=@Fecha\_Vencimiento\_Tarjeta,  
 Nombre\_Persona\_Asignada\_A\_Tarjeta=@Nombre\_Persona\_Asignada\_A\_Yarjeta,  
 Tipo\_TarjetaId=@Tipo\_TarjetaId  
 where TarjetaId= @TarjetaId;

END;

/\*Fin SP para tabla Tarjetas \*/

```

GO
/***** Object:  StoredProcedure [dbo].[SP_EditTipoContratos]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_EditTipoContratos]
    @TipoContratosId int,
    @Nombre_Tipo_Contrato varchar(25),
    @Descripcion varchar(100)
AS
Begin
    UPDATE tipo_contratos
    set Nombre_Tipo_Contrato=@Nombre_Tipo_Contrato, Descripcion=@Descripcion
    where Tipo_ContratoId= @TipoContratosId;
END;

/*Fin SP para tabla Tipo Contratos */

GO
/***** Object:  StoredProcedure [dbo].[SP_EditUnidadNegociosById]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_EditUnidadNegociosById]
    @UnidadNegocioId int,
    @Numero_Unidad_Negocio varchar(15),
    @Nombre_Unidad_Negocio varchar(15),
    @Nombre_Previo_Unidad_Negocio varchar(15)
AS
Begin
    UPDATE unidad_negocios
    set Numero_Unidad_Negocio=@Numero_Unidad_Negocio, Nombre_Unidad_Negocio=@Nombre_Unidad_Negocio,
    Nombre_Previo_Unidad_Negocio=@Nombre_Previo_Unidad_Negocio
    where Unidad_NegocioId=@UnidadNegocioId;
END;

/*Fin SP para tabla Unidad Negocios */
GO
/***** Object:  StoredProcedure [dbo].[SP_EditUser]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_EditUser]

    @UsuarioId int,
    @Nombre varchar(25),
    @Apellido varchar(25),
    @email varchar(50),
    @pass nvarchar(max),
    @Telefono varchar(20),
    @Codigo_Usuario varchar(15),

```

```

@username varchar(30),
@Direccion_Fotografia varchar (100),
@PerfilId int

AS
Begin
    OPEN SYMMETRIC KEY TestTableKey DECRYPTION
    BY CERTIFICATE EncryptTestCert

    UPDATE usuarios
    set Nombre=@Nombre, Apellido=@Apellido,
email=@email,pass='',pass2=ENCRYPTBYKEY(KEY_GUID('TestTableKey'),@pass),Telefono=@Telefono,Codigo_Us
uario=@Codigo_Usuario,username=@username,Direccion_Fotografia=@Direccion_Fotografia,
    PerfilId=@PerfilId
    where UsuarioId= @UsuarioId;

    update usuarios_perfiles_permisos set PerfilId=@PerfilId where UsuarioId = @UsuarioId

    CLOSE SYMMETRIC KEY TestTableKey;
END;

GO
/***** Object:  StoredProcedure [dbo].[SP_insertarAreaContratos]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_insertarAreaContratos]

    @Nombre_Area varchar(25)
AS
Begin
    insert into Area_Contratos(Nombre_Area)
    values (@Nombre_Area);
END;

/*Fin SP para tabla Area Contratos */

/* SP para tabla Contratos */

GO
/***** Object:  StoredProcedure [dbo].[SP_insertarBorradorContratos]    Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    CREATE procedure [dbo].[SP_insertarBorradorContratos]

    @Nombre_Contrato varchar(25),
    @Numero_Contrato varchar(15),
    @Numero_Servicio_Contrato varchar(15),

```

```

@Fecha_Inicio Date,
@Fecha_Finalizacion Date,
@Costo_Total numeric(18,2),
@Costo_Por_Cada_Pago numeric (18,2),
@Fecha_Renovacion_Pago Date,
@Cantidad_Licencias int,
@Garantia varchar(25),
@Fecha_Vencimiento_Garantia Date,
@Fecha_Creacion Date,
@Descripcion varchar(100),
@Direccion_Archivo varchar(100),
@Proveedor_ManufactureroId int,
@Proveedor_Distribuidor int,
@Usuario_AprobadorId int,
@Usuario_AdministradorId int,
@Tipo_ContratoId int,
@Metodo_PagoId int,
@Unidad_NegocioId int,
@Tipo_PagoId int,
@Orden_CompraId int,
@TarjetaId int,
@Area_ContratoId int,
@ProductoId int,
@RecordatorioId int,
@EstadoId int,
@Detalles_Producto varchar(max)

AS
Begin
    insert into
    Borradorcontratos(Nombre_Contrato2,Numero_Contrato2,Numero_Servicio_Contrato2,Fecha_Inicio2,Fecha_Fi
nalizacion2,Costo_Total2,Costo_Por_Cada_Pago2,

Fecha_Renovacion_Pago2,Cantidad_Licencias2,Garantia2,Fecha_Vencimiento_Garantia2,Fecha_Creacion2,Des
cripcion2,Direccion_Archivo2,Proveedor_ManufactureroId2,Proveedor_DistribuidorId2,

Usuario_AprobadorId2,Usuario_AdministradorId2,Tipo_ContratoId2,Metodo_PagoId2,Unidad_NegocioId2,Tipo
_PagoId2,Orden_CompraId2,TarjetaId2,Area_ContratoId2,ProductoId2,RecordatorioId2,EstadoId2,Detalles_
Producto2)
    values
    (@Nombre_Contrato,@Numero_Contrato,@Numero_Servicio_Contrato,@Fecha_Inicio,@Fecha_Finalizacion,@Cost
o_Total,@Costo_Por_Cada_Pago,@Fecha_Renovacion_Pago,

@Cantidad_Licencias,@Garantia,@Fecha_Vencimiento_Garantia,@Fecha_Creacion,@Descripcion,@Direccion_Ar
chivo,@Proveedor_ManufactureroId,@Proveedor_Distribuidor,@Usuario_AprobadorId,

@Usuario_AdministradorId,@Tipo_ContratoId,@Metodo_PagoId,@Unidad_NegocioId,@Tipo_PagoId,@Orden_Compr
aId,@TarjetaId,@Area_ContratoId,@ProductoId,@RecordatorioId,@EstadoId,@Detalles_Producto);
END;

GO
/***** Object:  StoredProcedure [dbo].[SP_insertarContratos]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE procedure [dbo].[SP_insertarContratos]

@Nombre_Contrato varchar(25),
@Numero_Contrato varchar(15),
@Numero_Servicio_Contrato varchar(15),
@Fecha_Inicio Date,
@Fecha_Finalizacion Date,
@Costo_Total numeric(18,2),
@Costo_Por_Cada_Pago numeric (18,2),
@Fecha_Renovacion_Pago Date,
@Cantidad_Licencias int,
@Garantia varchar(25),
@Fecha_Vencimiento_Garantia Date,
@Fecha_Creacion Date,
@Descripcion varchar(100),
@Direccion_Archivo varchar(100),
@Proveedor_ManufactureroId int,
@Proveedor_Distribuidor int,
@Usuario_AprobadorId int,
@Usuario_AdministradorId int,
@Tipo_ContratoId int,
@Metodo_PagoId int,
@Unidad_NegocioId int,
@Tipo_PagoId int,
@Orden_CompraId int,
@TarjetaId int,
@Area_ContratoId int,
@ProductoId int,
@RecordatorioId int,
@EstadoId int,
@Detalles_Producto varchar(max)

AS
Begin
    insert into
contratos(Nombre_Contrato,Numero_Contrato,Numero_Servicio_Contrato,Fecha_Inicio,Fecha_Finalizacion,C
osto_Total,Costo_Por_Cada_Pago,

Fecha_Renovacion_Pago,Cantidad_Licencias,Garantia,Fecha_Vencimiento_Garantia,Fecha_Creacion,Descripc
ion,Direccion_Archivo,Proveedor_ManufactureroId,Proveedor_DistribuidorId,

Usuario_AprobadorId,Usuario_AdministradorId,Tipo_ContratoId,Metodo_PagoId,Unidad_NegocioId,Tipo_Pago
Id,Orden_CompraId,TarjetaId,Area_ContratoId,ProductoId,RecordatorioId,EstadoId,Detalles_Producto)
    values
(@Nombre_Contrato,@Numero_Contrato,@Numero_Servicio_Contrato,@Fecha_Inicio,@Fecha_Finalizacion,@Cost
o_Total,@Costo_Por_Cada_Pago,@Fecha_Renovacion_Pago,

@Cantidad_Licencias,@Garantia,@Fecha_Vencimiento_Garantia,@Fecha_Creacion,@Descripcion,@Direccion_Ar
chivo,@Proveedor_ManufactureroId,@Proveedor_Distribuidor,@Usuario_AprobadorId,

@Usuario_AdministradorId,@Tipo_ContratoId,@Metodo_PagoId,@Unidad_NegocioId,@Tipo_PagoId,@Orden_Compr
aId,@TarjetaId,@Area_ContratoId,@ProductoId,@RecordatorioId,@EstadoId,@Detalles_Producto);
END;
GO
/***** Object:  StoredProcedure [dbo].[SP_insertarEstados]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO

```

```

SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_insertarEstados]

@Descripcion varchar(25)
As
begin
insert into Estados( Descripcion)
values (@Descripcion);
End;
/*Fin SP para tabla Estados */

/* SP para tabla Metodo Pagos */

GO
/***** Object:  StoredProcedure [dbo].[SP_insertarMetodoPagos]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_insertarMetodoPagos]

@Nombre_Metodo_Pago varchar(25)

AS
Begin
insert into Metodo_Pagos(Nombre_Metodo_Pago)
values (@Nombre_Metodo_Pago);
END;

/*Fin SP para tabla Metodo Pagos */

/* SP para tabla Orden Compras */

GO
/***** Object:  StoredProcedure [dbo].[SP_insertarOrdenCompras]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_insertarOrdenCompras]
@Fecha_Ultima_Orden_Compra Date,
@Ultimo_Numero_Factura varchar(15),
@Codigo_Presupuestario_2016 varchar(15),
@Monto_Previo varchar(25),
@Direccion_Archivo varchar(100)

AS
Begin
insert into orden_compras
(Fecha_Ultima_Orden_Compra,Ultimo_Numero_Factura,Codigo_Presupuestario2016,Monto_Previo,Direccion_Ar
chivo)

```

```

        values
        (@Fecha_Ultima_Orden_Compra,@Ultimo_Numero_Factura,@Codigo_Presupuestario_2016,@Monto_Previo,@Direcc
        ion_Archivo)
    END;

/*Fin SP para tabla Orden Compras */

/* SP para tabla Perfiles */

GO
/***** Object:  StoredProcedure [dbo].[SP_insertarPerfiles]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_insertarPerfiles]

        @Nombre_Perfil varchar(25),
        @Descripcion varchar(100)
AS
Begin
    insert into perfiles(Nombre_Perfil,Descripcion)
    values (@Nombre_Perfil,@Descripcion);
END;

/*Fin SP para tabla Perfiles */

/* SP para tabla Productos */

GO
/***** Object:  StoredProcedure [dbo].[SP_insertarProductos]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_insertarProductos]

        @Nombre_Producto varchar(25),
        @Precio numeric(18,2),
        @Descripcion varchar(100),
        @Direccion_Fotografia varchar(100),
        @Proveedor_ManufactureroId int,
        @Proveedor_DistribuidorId int
AS
Begin
    insert into
    productos(Nombre_Producto,Precio,Descripcion,Direccion_Fotografia,Proveedor_ManufactureroId,Proveedo
    r_DistribuidorId)
    values
    (@Nombre_Producto,@Precio,@Descripcion,@Direccion_Fotografia,@Proveedor_ManufactureroId,@Proveedor_D
    istribuidorId);
END;

/*Fin SP para tabla Productos */

```

```
/* SP para tabla Proveedores */
```

```
GO
/***** Object:  StoredProcedure [dbo].[SP_insertarProveedores]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_insertarProveedores]

@Nombre_Proveedor varchar(25),
@Correo_Proveedor varchar(50),
@Nombre_Contacto varchar(25),
@Telefono_Contacto varchar(20),
@Ubicacion varchar(50),
@Descripcion varchar(100),
@Tipo_ProveedorId int
AS
Begin
insert into
proveedores(Nombre_Proveedor, Correo_Proveedor, Nombre_Contacto, Telefono_Contacto, Ubicacion, Descripcio
n, Tipo_ProveedorId)
values
(@Nombre_Proveedor, @Correo_Proveedor, @Nombre_Contacto, @Telefono_Contacto, @Ubicacion, @Descripcion, @Ti
po_ProveedorId);
END;

/*Fin SP para tabla Proveedores */

/* SP para tabla Tarjetas */
```

```
GO
/***** Object:  StoredProcedure [dbo].[SP_insertarTarjetas]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_insertarTarjetas]

@Ultimos_Digitos_Tarjeta varchar(4),
@Fecha_Vencimiento_Tarjeta Date,
@Nombre_Persona_Asignada_A_Yarjeta varchar(25),
@Tipo_TarjetaId int
AS
Begin
```



```

    insert into
Tarjetas(Ultimos_Digitos_Tarjeta,Fecha_Vencimiento_Tarjeta,Nombre_Persona_Asignada_A_Tarjeta,Tipo_Ta
rjetaId)
    values
(@Ultimos_Digitos_Tarjeta,@Fecha_Vencimiento_Tarjeta,@Nombre_Persona_Asignada_A_Yarjeta,@Tipo_Tarjet
aId);
END;

/*Fin SP para tabla Tarjetas */

/* SP para tabla Tipo Contratos */

GO
/***** Object:  StoredProcedure [dbo].[SP_insertarTipoContratos]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_insertarTipoContratos]
@Nombre_Tipo_Contrato varchar(25),
@Descripcion varchar(100)
AS
Begin
    insert into Tipo_Contratos(Nombre_Tipo_Contrato,Descripcion)
    values (@Nombre_Tipo_Contrato,@Descripcion);
END;

/*Fin SP para tabla Tipo Contratos */

/* SP para tabla Tipo Pagos */

GO
/***** Object:  StoredProcedure [dbo].[SP_insertarTipoPagos]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_insertarTipoPagos]
@Nombre_Tipo_Pago varchar(25)
AS
Begin
    insert into tipo_pagos(Nombre_Tipo_Pago)
    values (@Nombre_Tipo_Pago);
END;

/*Fin SP para tabla Tipo Pagos */

/* SP para tabla Unidad Negocios */

```

```

GO
/***** Object:  StoredProcedure [dbo].[SP_insertarUnidadNegocios]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_insertarUnidadNegocios]
@Numero_Unidad_Negocio varchar(15),
@Nombre_Unidad_Negocio varchar(15),
@Nombre_Previo_Unidad_Negocio varchar(15)
AS
Begin
insert into
Unidad_Negocios(Numero_Unidad_Negocio,Nombre_Unidad_Negocio,Nombre_Previo_Unidad_Negocio)
values (@Numero_Unidad_Negocio,@Nombre_Unidad_Negocio,@Nombre_Previo_Unidad_Negocio);
END;

/*Fin SP para tabla Unidad Negocios */

/* SP para tabla Usuarios */

GO
/***** Object:  StoredProcedure [dbo].[SP_insertarUsuarios]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_insertarUsuarios]
@Nombre varchar(25),
@Apellido varchar(25),
@email varchar(50),
@pass nvarchar(max),
@Telefono varchar(20),
@Codigo_Usuario varchar(15),
@Username varchar (30),
@Fecha_Creacion Datetime,
@Direccion_Fotografia varchar (100),
@PerfilId int
AS
Begin
OPEN SYMMETRIC KEY TestTableKey DECRYPTION
BY CERTIFICATE EncryptTestCert

insert into
Usuarios(Nombre,Apellido,email,pass,pass2,Telefono,Codigo_Usuario,username,Fecha_Creacion,Direccion_
Fotografia,PerfilId)
values
(@Nombre,@Apellido,@email, '', ENCRYPTBYKEY(KEY_GUID('TestTableKey'),@pass),@Telefono,@Codigo_Usuario,
@Username,@Fecha_Creacion,@Direccion_Fotografia,@PerfilId);

CLOSE SYMMETRIC KEY TestTableKey;

```

```

declare @idUser int
set @idUser = SCOPE_IDENTITY()

if(@PerfilId=1)
    insert into usuarios_perfiles_permisos(UsuarioId,PerfilId,PermisoId) values
(@idUser,@PerfilId,1);

else
    if(@PerfilId =2 or @PerfilId=3 or @PerfilId =4)
        insert into usuarios_perfiles_permisos(UsuarioId,PerfilId,PermisoId) values
(@idUser,@PerfilId,2);

    else
        if(@PerfilId =6 or @PerfilId=7 or @PerfilId=8)
            insert into usuarios_perfiles_permisos(UsuarioId,PerfilId,PermisoId) values
(@idUser,@PerfilId,3);

        else
            if(@PerfilId =5)
                insert into usuarios_perfiles_permisos(UsuarioId,PerfilId,PermisoId) values
(@idUser,@PerfilId,4);
            END;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectAdministrator]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectAdministrator]
AS
Begin
    SELECT  UsuarioId,Nombre,Apellido  FROM usuarios where (PerfilId=2 or PerfilId=3 or PerfilId=4)
and UsuarioId !=0;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectAdminUserPermissions]    Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_selectAdminUserPermissions]
@usuarioId int
AS
Begin
    SELECT  UsuarioId, PerfilId, PermisoId, Usuario_Perfil_PermisoId
FROM      usuarios_perfiles_permisos
    where UsuarioId = @usuarioId ;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectAllBusinessunit]    Script Date: 7/20/2016 2:01:57
PM *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectAllBusinessunit]
AS
Begin
    SELECT * FROM unidad_negocios where Unidad_NegocioId !=0;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectAllCards]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectAllCards]
AS
Begin

SELECT    dbo.tarjetas.TarjetaId, dbo.tarjetas.Ultimos_Digitos_Tarjeta,
dbo.tarjetas.Fecha_Vencimiento_Tarjeta, dbo.tarjetas.Nombre_Persona_Asignada_A_Tarjeta,
dbo.tipo_tarjetas.Nombre_Tarjeta
FROM      dbo.tarjetas INNER JOIN
          dbo.tipo_tarjetas ON dbo.tarjetas.Tipo_TarjetaId = dbo.tipo_tarjetas.Tipo_TarjetaId
          where TarjetaId !=0;

End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectAllContractArea]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
Create procedure [dbo].[SP_selectAllContractArea]
AS
Begin
    SELECT * FROM area_contratos;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectAllDetailsContract]    Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectAllDetailsContract]
@contratoId int
AS
Begin

SELECT    contratos.ContratoId, contratos.Nombre_Contrato, contratos.Numero_Contrato,
contratos.Numero_Servicio_Contrato, contratos.Fecha_Inicio, contratos.Fecha_Finalizacion,
contratos.Costo_Total,

```

```

        contratos.Costo_Por_Cada_Pago, contratos.Fecha_Renovacion_Pago,
        contratos.Cantidad_Licencias, contratos.Garantia, contratos.Fecha_Vencimiento_Garantia,
        contratos.Proveedor_ManufactureroId,
        contratos.Proveedor_DistribuidorId, contratos.Usuario_AprobadorId,
        contratos.Usuario_AdministradorId, unidad_negocios.Nombre_Unidad_Negocio, estados.Descripcion,
        metodo_pagos.Nombre_Metodo_Pago,
        tarjetas.Ultimos_Digitos_Tarjeta, tarjetas.Fecha_Vencimiento_Tarjeta,
        tarjetas.Nombre_Persona_Asignada_A_Tarjeta, orden_compras.Codigo_Presupuestario2016,
        productos.Nombre_Producto,
        contratos.ProductoId, tipo_contratos.Nombre_Tipo_Contrato, tipo_pagos.Nombre_Tipo_Pago,
        contratos.Direccion_Archivo, contratos.Orden_CompraId,
        contratos.Fecha_Creacion, contratos.Detalles_Producto

```

```

FROM    contratos INNER JOIN
        estados ON contratos.EstadoId = estados.EstadoId INNER JOIN
        metodo_pagos ON contratos.Metodo_PagoId = metodo_pagos.Metodo_PagoId INNER JOIN
        orden_compras ON contratos.Orden_CompraId = orden_compras.Orden_CompraId INNER JOIN
                productos ON contratos.ProductoId = productos.ProductoId INNER JOIN
                tarjetas ON contratos.TarjetaId = tarjetas.TarjetaId INNER JOIN
                tipo_contratos ON contratos.Tipo_ContratoId =
tipo_contratos.Tipo_ContratoId INNER JOIN
                tipo_pagos ON contratos.Tipo_PagoId = tipo_pagos.Tipo_PagoId INNER JOIN
                unidad_negocios ON contratos.Unidad_NegocioId =
unidad_negocios.Unidad_NegocioId

```

```
where ContratoId=@contratoId
```

```
End;
```

```
GO
/***** Object: StoredProcedure [dbo].[SP_selectAllMethodPayment]    Script Date: 7/20/2016 2:01:57
PM *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
Create procedure [dbo].[SP_selectAllMethodPayment]
```

```
AS
```

```
Begin
```

```
SELECT * FROM metodo_pagos;
```

```
End;
```

```
GO
/***** Object: StoredProcedure [dbo].[SP_selectAllProducts]    Script Date: 7/20/2016 2:01:57 PM
*****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE procedure [dbo].[SP_selectAllProducts]
```

```
AS
```

```
Begin
```

```
SELECT * FROM productos where ProductoId !=0;
```

```
End;
```

```
GO
/***** Object: StoredProcedure [dbo].[SP_selectAllProvider]    Script Date: 7/20/2016 2:01:57 PM
*****/
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectAllProvider]
AS
Begin

SELECT  dbo.proveedores.ProveedorId, dbo.proveedores.Nombre_Proveedor,
dbo.proveedores.Correo_Proveedor, dbo.proveedores.Nombre_Contacto,
dbo.proveedores.Telefono_Contacto, dbo.proveedores.Ubicacion,
        dbo.proveedores.Descripcion, dbo.tipo_proveedores.Descripcion AS Expr1
FROM    dbo.proveedores INNER JOIN
        dbo.tipo_proveedores ON dbo.proveedores.Tipo_ProveedorId =
dbo.tipo_proveedores.Tipo_ProveedorId where ProveedorId !=0;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectAllPurchaseOrder]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectAllPurchaseOrder]
AS
Begin
SELECT * FROM orden_compras where Orden_CompraId != 0;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectAllTypeContract]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_selectAllTypeContract]
AS
Begin
SELECT * FROM tipo_contratos;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectAllTypeProvider]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
Create procedure [dbo].[SP_selectAllTypeProvider]
AS
Begin
SELECT * FROM tipo_proveedores;
End;
```

```

GO
/***** Object:  StoredProcedure [dbo].[SP_selectAllUser]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectAllUser]
AS
Begin

SELECT dbo.usuarios.UsuarioId, dbo.usuarios.Nombre, dbo.usuarios.Apellido, dbo.usuarios.email,
dbo.usuarios.Telefono, dbo.usuarios.Codigo_Usuario, dbo.usuarios.username,
dbo.perfiles.Nombre_Perfil
FROM   dbo.perfiles INNER JOIN
        dbo.usuarios ON dbo.perfiles.PerfilId = dbo.usuarios.PerfilId  where UsuarioId != 0  and
UsuarioId !=1;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectApprover]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectApprover]
AS
Begin
SELECT UsuarioId,Nombre , Apellido FROM usuarios where PerfilId=5 and UsuarioId !=1;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectBusinessUnit]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_selectBusinessUnit]
AS
Begin
SELECT Nombre_Unidad_Negocio FROM unidad_negocios;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectCard]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectCard]
AS
Begin
SELECT TarjetaId,tipo_tarjetas.Nombre_Tarjeta, Ultimos_Digitos_Tarjeta
FROM   tarjetas INNER JOIN tipo_tarjetas ON dbo.tarjetas.Tipo_TarjetaId =
dbo.tipo_tarjetas.Tipo_TarjetaId

```

```

where tarjetaId != 0
End;

```

```

GO
/***** Object:  StoredProcedure [dbo].[SP_selectContract]    Script Date: 7/20/2016 2:01:57 PM
*****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE procedure [dbo].[SP_selectContract]

```

```

AS

```

```

Begin

```

```

SELECT

```

```

    dbo.contratos.ContratoId, dbo.contratos.Nombre_Contrato, dbo.contratos.Numero_Contrato,
    dbo.contratos.Costo_Por_Cada_Pago,
    dbo.tipo_pagos.Nombre_Tipo_Pago, dbo.contratos.Fecha_Renovacion_Pago, dbo.estados.Descripcion,
    dbo.tipo_contratos.Nombre_Tipo_Contrato

```

```

FROM

```

```

    dbo.area_contratos INNER JOIN
    dbo.contratos ON dbo.area_contratos.Area_ContratoId = dbo.contratos.Area_ContratoId INNER JOIN
    dbo.estados ON dbo.contratos.EstadoId = dbo.estados.EstadoId INNER JOIN
    dbo.tipo_contratos ON dbo.contratos.Tipo_ContratoId = dbo.tipo_contratos.Tipo_ContratoId INNER

```

```

JOIN

```

```

    dbo.tipo_pagos ON dbo.contratos.Tipo_PagoId = dbo.tipo_pagos.Tipo_PagoId

```

```

End;

```

```

GO
/***** Object:  StoredProcedure [dbo].[SP_selectContractArea]    Script Date: 7/20/2016 2:01:57 PM
*****/

```

```

SET ANSI_NULLS ON

```

```

GO

```

```

SET QUOTED_IDENTIFIER ON

```

```

GO

```

```

create procedure [dbo].[SP_selectContractArea]

```

```

AS

```

```

Begin

```

```

    SELECT Nombre_Area FROM area_contratos;

```

```

End;

```

```

GO
/***** Object:  StoredProcedure [dbo].[SP_selectContractAreaById]    Script Date: 7/20/2016 2:01:57 PM
*****/

```

```

SET ANSI_NULLS ON

```

```

GO

```

```

SET QUOTED_IDENTIFIER ON

```

```

GO

```

```

Create procedure [dbo].[SP_selectContractAreaById]

```

```

    @Area_ContratoId int

```

```

AS

```

```

Begin

```

```

    SELECT * FROM area_contratos
    where Area_ContratoId =@Area_ContratoId;

```

```

End;

```



```

GO
/***** Object:  StoredProcedure [dbo].[SP_selectContractByArea]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_selectContractByArea]
        @CodigoArea int
AS
Begin

SELECT
    dbo.contratos.ContratoId, dbo.contratos.Nombre_Contrato, dbo.contratos.Numero_Contrato,
    dbo.contratos.Costo_Por_Cada_Pago,
    dbo.tipo_pagos.Nombre_Tipo_Pago, dbo.contratos.Fecha_Renovacion_Pago, dbo.estados.Descripcion,
    dbo.tipo_contratos.Nombre_Tipo_Contrato

FROM
    dbo.area_contratos INNER JOIN
    dbo.contratos ON dbo.area_contratos.Area_ContratoId = dbo.contratos.Area_ContratoId INNER JOIN
    dbo.estados ON dbo.contratos.EstadoId = dbo.estados.EstadoId INNER JOIN
    dbo.tipo_contratos ON dbo.contratos.Tipo_ContratoId = dbo.tipo_contratos.Tipo_ContratoId INNER
JOIN
    dbo.tipo_pagos ON dbo.contratos.Tipo_PagoId = dbo.tipo_pagos.Tipo_PagoId

    where dbo.contratos.Area_ContratoId = @CodigoArea;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_SelectContractForEdit]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_SelectContractForEdit]
@ContratoId int
as

SELECT      dbo.contratos.*, dbo.productos.Nombre_Producto
FROM        dbo.contratos INNER JOIN
            dbo.productos ON dbo.contratos.ProductoId = dbo.productos.ProductoId

where ContratoId= @contratoId;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectDraftContract]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    CREATE procedure [dbo].[SP_selectDraftContract]
AS
Begin
SELECT

```

```

        dbo.Borradorcontratos.ContratoId2, dbo.Borradorcontratos.Nombre_Contrato2,
        dbo.Borradorcontratos.Numero_Contrato2,
            dbo.Borradorcontratos.Numero_Servicio_Contrato2,
        dbo.Borradorcontratos.Fecha_Inicio2,
            dbo.Borradorcontratos.Fecha_Finalizacion2, dbo.Borradorcontratos.Costo_Total2,
        dbo.Borradorcontratos.Costo_Por_Cada_Pago2

```

```

FROM          dbo.Borradorcontratos
End;

```

```

GO
/***** Object:  StoredProcedure [dbo].[SP_selectDraftContractByArea]    Script Date: 7/20/2016
2:01:57 PM *****/

```

```

SET ANSI_NULLS ON

```

```

GO

```

```

SET QUOTED_IDENTIFIER ON

```

```

GO

```

```

CREATE procedure [dbo].[SP_selectDraftContractByArea]
    @CodigoArea int

```

```

AS

```

```

Begin

```

```

SELECT

```

```

        dbo.Borradorcontratos.ContratoId2, dbo.Borradorcontratos.Nombre_Contrato2,
        dbo.Borradorcontratos.Numero_Contrato2,
            dbo.Borradorcontratos.Numero_Servicio_Contrato2,
        dbo.Borradorcontratos.Fecha_Inicio2,
            dbo.Borradorcontratos.Fecha_Finalizacion2, dbo.Borradorcontratos.Costo_Total2,
        dbo.Borradorcontratos.Costo_Por_Cada_Pago2

```

```

FROM          dbo.Borradorcontratos

```

```

        where dbo.Borradorcontratos.Area_ContratoId2 = @CodigoArea;

```

```

End;

```

```

GO
/***** Object:  StoredProcedure [dbo].[SP_SelectDraftContractForEdit]    Script Date: 7/20/2016
2:01:57 PM *****/

```

```

SET ANSI_NULLS ON

```

```

GO

```

```

SET QUOTED_IDENTIFIER ON

```

```

GO

```

```

create procedure [dbo].[SP_SelectDraftContractForEdit]
    @ContratoId int

```

```

as

```

```

SELECT  dbo.Borradorcontratos.*, dbo.productos.Nombre_Producto
FROM    dbo.Borradorcontratos INNER JOIN
        dbo.productos ON dbo.Borradorcontratos.ProductoId2 = dbo.productos.ProductoId

```

```

where ContratoId2= @ContratoId;

```

```

GO
/***** Object:  StoredProcedure [dbo].[SP_selectMethodPayment]    Script Date: 7/20/2016 2:01:57 PM
*****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_selectMethodPayment]
AS
Begin
    SELECT Nombre_Metodo_Pago FROM metodo_pagos;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectMethodPaymentById]    Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_selectMethodPaymentById]
@MetodoPagoId int
AS
Begin
    SELECT * FROM metodo_pagos
    where Metodo_PagoId= @MetodoPagoId;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_SelectOrdenComprasById]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_SelectOrdenComprasById]
@OrdenCompraId int
AS
Begin
    select * from orden_compras where Orden_CompraId=@OrdenCompraId
END;

/*Fin SP para tabla Orden Compras */

/* SP para tabla Perfiles */

GO
/***** Object: StoredProcedure [dbo].[SP_selectPass]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectPass]
@email varchar(50)
AS
Begin

```

```

-- select top 1 pass from usuarios where email = @email;
OPEN SYMMETRIC KEY TestTableKey DECRYPTION
BY CERTIFICATE EncryptTestCert

select top 1 CONVERT(nVARCHAR(max),DECRYPTBYKEY(pass2)) from usuarios where email = @email;

CLOSE SYMMETRIC KEY TestTableKey;

End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectProductById]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_selectProductById]
@ProductId int
AS
Begin
SELECT * FROM productos where ProductoId=@ProductId;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectProductByName]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_selectProductByName]
@ProductName varchar(25)
AS
Begin
SELECT * FROM productos where Nombre_Producto=@ProductName;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectProductByProviderId]    Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectProductByProviderId]
@ProveedorId int
AS
Begin
select * from productos where (Proveedor_ManufactureroId=@ProveedorId or
Proveedor_DistribuidorId=@ProveedorId) and ProductoId !=0;
End;

```

```
GO
/***** Object: StoredProcedure [dbo].[SP_selectProductsAccordingSuppliers]    Script Date:
7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectProductsAccordingSuppliers]
    @ProviderManufacturing int,
    @ProviderVendor int
AS
Begin

SELECT
    ProductoId,Nombre_Producto

FROM
    productos
    where Proveedor_ManufactureroId = @ProviderManufacturing and
Proveedor_DistribuidorId=@ProviderVendor and ProductoId !=0;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectProfiles]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_selectProfiles]
AS
Begin
    SELECT Nombre_Perfil FROM perfiles;

END;

GO
/***** Object: StoredProcedure [dbo].[SP_selectProviderById]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectProviderById]
    @ProveedorId int
AS
Begin
    SELECT * FROM proveedores where ProveedorId=@ProveedorId;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectProviderManufacturing]    Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```

CREATE procedure [dbo].[SP_selectProviderManufacturing]
AS
Begin
    SELECT ProveedorId, Nombre_Proveedor FROM proveedores where Tipo_ProveedorId=1 and
ProveedorId !=0;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectProviderVender]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectProviderVender]
AS
Begin
    SELECT ProveedorId, Nombre_Proveedor FROM proveedores where Tipo_ProveedorId=2;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectPurchaseOrder]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectPurchaseOrder]
AS
Begin
    SELECT Orden_CompraId, Codigo_Presupuestario2016 FROM orden_compras where Orden_CompraId!=0;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectState]    Script Date: 7/20/2016 2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_selectState]
AS
Begin
    SELECT Descripcion FROM estados;
End;

GO
/***** Object: StoredProcedure [dbo].[SP_selectTarjetasById]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_selectTarjetasById]
@TarjetaId int
AS

```

```
Begin
    select * from tarjetas
    where TarjetaId= @TarjetaId;
END;

/*Fin SP para tabla Tarjetas */

GO
/***** Object:  StoredProcedure [dbo].[SP_selectTypeCard]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_selectTypeCard]
AS
Begin
    SELECT Nombre_Tarjeta FROM tipo_tarjetas;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectTypeContract]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_selectTypeContract]
AS
Begin
    SELECT Nombre_Tipo_Contrato FROM tipo_contratos;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectTypeContractById]    Script Date: 7/20/2016 2:01:57
PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_selectTypeContractById]
    @TipoContratoId int
AS
Begin
    SELECT * FROM tipo_contratos where Tipo_ContratoId=@TipoContratoId;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectTypePayment]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
    create procedure [dbo].[SP_selectTypePayment]
AS
Begin
```

```

        SELECT Nombre_Tipo_Pago FROM tipo_pagos;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectTypeProvider]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_selectTypeProvider]
AS
Begin
    SELECT Descripcion FROM tipo_proveedores;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_SelectUnidadNegociosById]    Script Date: 7/20/2016
2:01:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_SelectUnidadNegociosById]
    @UnidadNegocioId int
AS
Begin
    select * from unidad_negocios
    where Unidad_NegocioId=@UnidadNegocioId;
END;

/*Fin SP para tabla Unidad Negocios */
GO
/***** Object:  StoredProcedure [dbo].[SP_selectUserById]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectUserById]
    @UsuarioId int
AS
Begin
    OPEN SYMMETRIC KEY TestTableKey DECRYPTION
    BY CERTIFICATE EncryptTestCert

    SELECT UsuarioId,Nombre,Apellido,email,CONVERT(nVARCHAR(max),DECRYPTBYKEY(pass2)) as
    pass,Telefono,Codigo_Usuario,username,
    Fecha_Creacion,Direccion_Fotografia,PerfilId
    FROM usuarios
    where UsuarioId=@UsuarioId;

    CLOSE SYMMETRIC KEY TestTableKey;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectUserId]    Script Date: 7/20/2016 2:01:57 PM
*****/

```



```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[SP_selectUserId]
    @username varchar(250),
    @pass varchar(250)
AS
Begin

    OPEN SYMMETRIC KEY TestTableKey DECRYPTION
    BY CERTIFICATE EncryptTestCert

    SELECT UsuarioId FROM usuarios
    where username = @username and CONVERT(nVARCHAR(max),DECRYPTBYKEY(pass2))=@pass

    CLOSE SYMMETRIC KEY TestTableKey;
End;

GO
/***** Object:  StoredProcedure [dbo].[SP_selectWeatherAlerts]    Script Date: 7/20/2016 2:01:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[SP_selectWeatherAlerts]
AS
Begin
    SELECT Descripcion FROM recordatorios;
End;

GO
USE [master]
GO
ALTER DATABASE [sistema_alertas] SET READ_WRITE
GO
```

## 7. DISEÑO DE PANTALLAS

The image displays a user interface for Laureate International Universities, featuring a login and password recovery section. The top half has an orange background with a white circle on the left containing the text "HERE FOR GOOD". In the center is a white login box with a blue user icon, a username input field labeled "Enter your username:", a password input field with a yellow background and masked characters, a green "Login" button, and a link "¿I can not access your account?". To the right is the Laureate International Universities logo. Below this is a dark grey header bar with the Laureate logo. The bottom half has a light grey background with a white wave pattern. On the left, there is a "BACK LOGIN" button, a "Recover Password" section with an "Email \*" input field, and a "SEND MAIL" button. On the right, a teal box titled "Recover Password" contains instructions: "Your password will be sent to the email with which you registered in the system" and "If the mail does not appear in your inbox please check your spam tray can post your customer you have moved to this folder".

HERE FOR GOOD

LAUREATE INTERNATIONAL UNIVERSITIES

LAUREATE INTERNATIONAL UNIVERSITIES

BACK LOGIN

Recover Password


Email \*

SEND MAIL

Recover Password

Your password will be sent to the email with which you registered in the system

If the mail does not appear in your inbox please check your spam tray can post your customer you have moved to this folder


Manage Site Log Out

CONTRACTS
REPORTS


Search:

Name of Contract	Contract Number	Type of contract	Cost for each payment	Payment type	Renewal Payment Date	State	Edition
Amazon Web Services	746822605337	Maintenance	2400.000	Monthly	8/3/2016	Active	<a href="#">Edit</a> <a href="#">Remove</a>
Google Apps	7101-3889-8419-	License	\$25.00	Monthly	8/3/2016	Active	<a href="#">Edit</a> <a href="#">Remove</a>
Kaltura		License	236790.000	Annual	3/31/2017	Active	<a href="#">Edit</a> <a href="#">Remove</a>

Showing 1 to 3 of 3 entries
Previous
1
Next

### Contracts

The contract management system allows you to manage contracts with external suppliers. Three types of contracts are allowed: leasing, maintenance, and software license.


Log Out

#### General description:

Name: Amazon Web Services

State Contract: Active

Type of contract: Maintenance

Due date: 12/29/2016

Payment method: Credit Card

Total cost(\$): \$28,800.000

Payment type: Monthly

Cost Of Payment(\$): \$2,400.000

Renewal Payment Date: 8/3/2016

Creation Date in the system: 7/20/2016

#### Details:

Contract Number: 746822605337      Name Contract: Amazon Web Services

Service Contract Number: 746822605337      Business Unit: HND10

Contract type: Maintenance      State Contract: Active

Supplier Manufacturing: Amazon      Seller supplier: Amazon

Approver: Aprobador Aprobador      Administrator: Sara Castillo

Total cost(\$): \$28,800.000      Payment type: Monthly

Cost Of Payment(\$): \$2,400.000      Renewal Date Payment: 8/3/2016

Warranty: N/A      Guarantee Maturity Date: N/A

Number Licenses: N/A      Budgetary code Purchase Order: N/A

Date Start Contract: 7/29/2016      Date End Contract: 12/29/2016

Payment method: Credit Card      Last Digit Card: 2009

Name appointee to card: Silvia Coello      Card Expiration Date: 3/31/2020

Attached files:      Description:

#### Associated Products A This Agreement

S3

EC2


RDS

Data Transfer

Data Pipeline

Cloud Front

[BACK TO CONTRACTS](#)

 Log Out

General Description

Name Supplier: Amazon

Mail Provider: accounts@amazon.com

Contact Supplier: Online

Phone Contact:


Location Provider

Description: Online Services from Amazon

BACK TO CONTRACTS

List of products from this supplier:

S3
EC2
RDS
Data Transfer
Data Pipeline
Cloud Front

 Log Out

General description

Name: Aprobador Aprobador

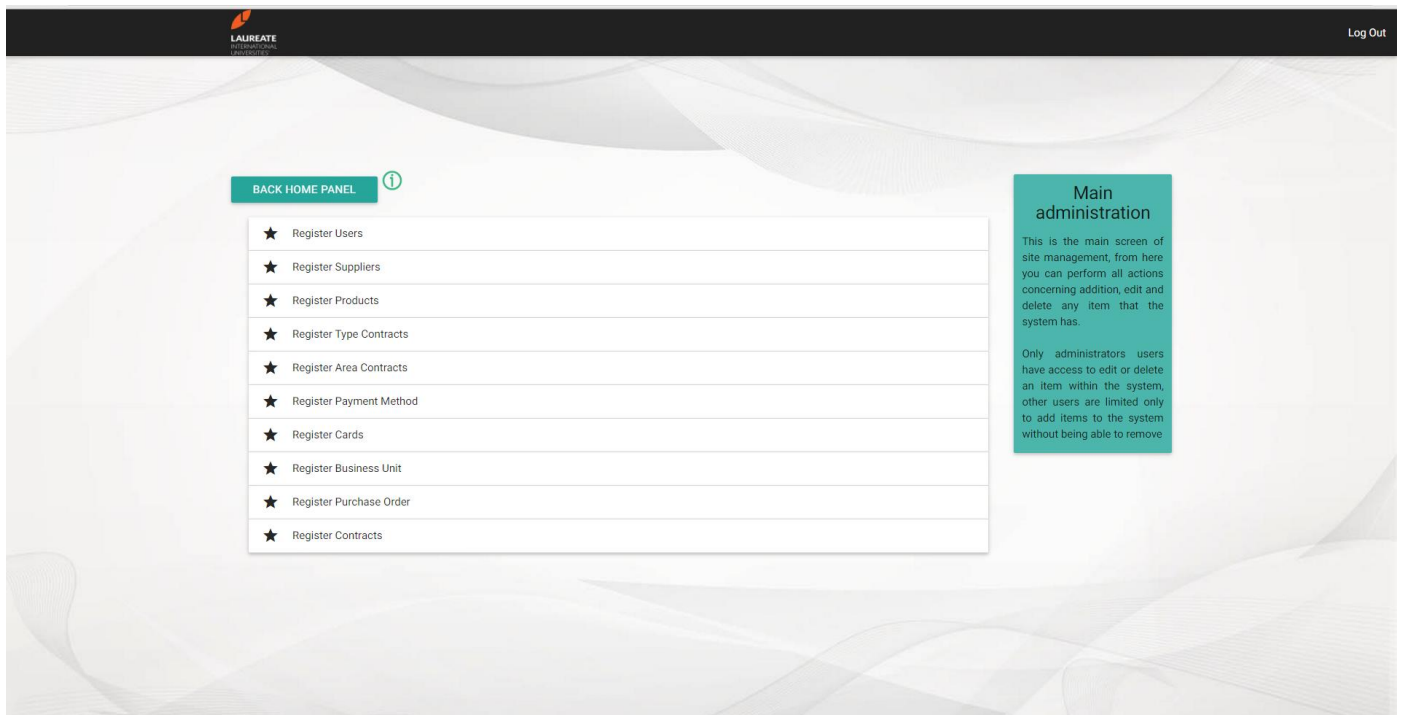
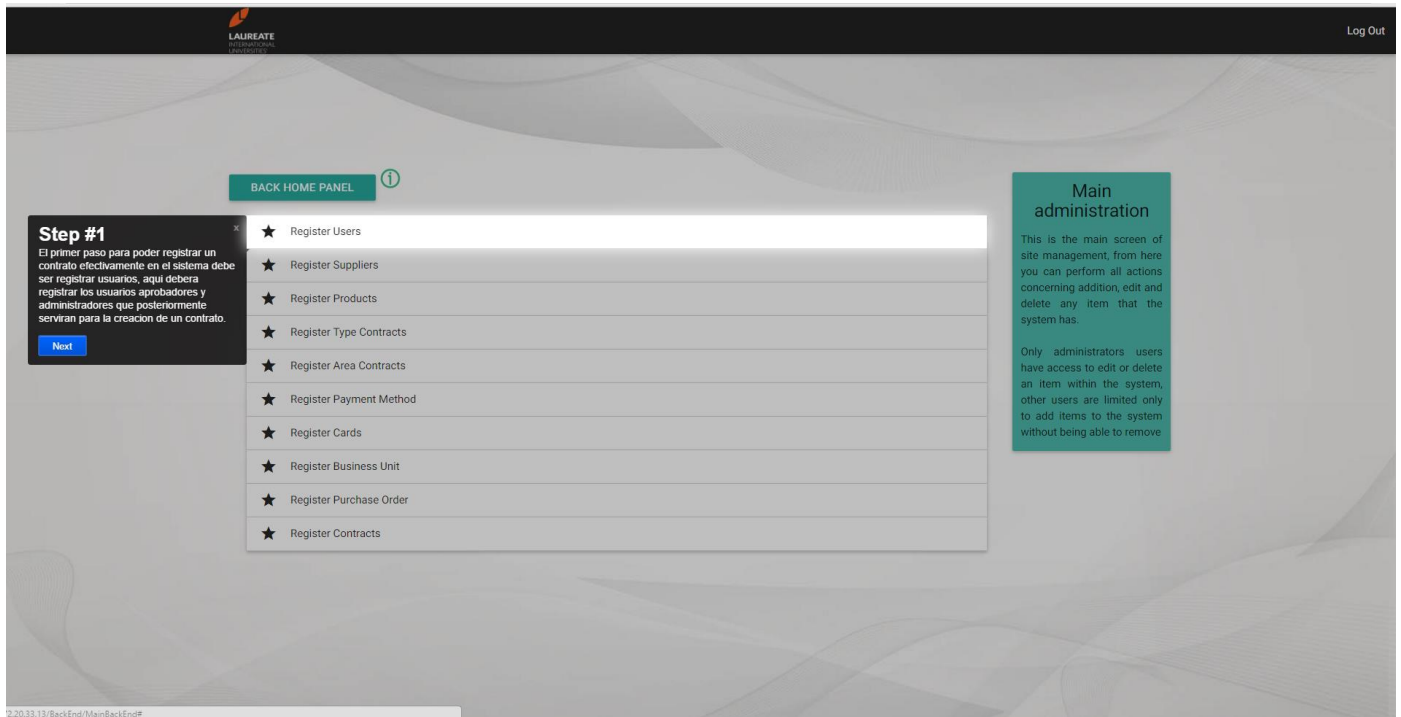
Email: cturcioscolindres38@yahoo.com


Phone Contact:

User code:

BACK TO CONTRACTS

Aprobador Aprobador




Log Out

BACK PANEL ADMINISTRATION

Creating a new user

Name \*

Last name \*

Phone

Mail \* laureate@laureate.com

Position within the system \*

Select

User code

Username \*

Password \*

Enter Password again \*

IMAGE

Upload a picture displayed on the user's profile


RETURN TO DRAFTS

SAVE

User registration

The user creation module allows you to add a new user to the system. This will be the first step for you to use the system, you won't be able to access the site without a creating a user.

All users created in the system should have a specific role, which serve to determine what actions they can or can't perform.


Log Out

BACK PANEL ADMINISTRATION

Creating new provider

Name Provider \*

Mail Provider \*

Type Provider \*

Select

Primary Contact Name Supplier

Phone Contact

Location Provider

Description supplier location

Description

A brief description about the provider


RETURN TO DRAFTS

SAVE

Supplier Registration

The vendor management module allows you to register new vendors into the system in order to maintain an order of all external suppliers who provide a service or sell some product to the company.

Vendors are divided into two types: 1) Manufacturing Suppliers 2) Distributing Suppliers.

LAUREATE  
INTERNATIONAL  
ACADEMY

Log Out

BACK PANEL ADMINISTRATION

Creation of new products in the system

Product name \*

Single Price (\$) \*

Supplier that manufactures the product \*

Provider that distributes the product \*

Seleccione ▼

Select ▼

Description

A brief description about the product

IMAGE

Upload a photo that will appear in the product profile


CANCEL

SAVE

Product Registration

The management module Product Registration allows you to add new products to the system, those products registered in the system are dependent on the suppliers who have previously been registered.

To register a new product you must specify who the manufacturing supplier and distributing provider are.

LAUREATE  
INTERNATIONAL  
ACADEMY

Log Out

BACK PANEL ADMINISTRATION

Creation of new types of contract

Name Type Contract \*

Description

A brief description about the product

CANCEL


SAVE

Registration Contract Types

The management record contract module allows you to register a new type of contract in the system. By default the system handles 3 types of contracts.

1) License 2) Lease 3) Maintenance. But you can add a new type of contract if needed.



LAUREATE  
INTERNATIONAL  
EDUCATION

Log Out

BACK PANEL ADMINISTRATION

Creating a new area of contract


Area Name \*

CANCEL

SAVE

**Registration Area Contracts**

The contract registration module area lets you add a new area of contracts to the system. Creating new areas of contracts will allow the user to view only contracts pertaining to their specific area.

LAUREATE  
INTERNATIONAL  
EDUCATION

Log Out

BACK PANEL ADMINISTRATION

Creating a new payment method

Name of payment method \*

CANCEL


SAVE

**Registration Payment Method**

The registration module payment method allows you to add a new payment method to the system.

The payment methods that the system defines by default are listed below: 1) Cash 2) Cheque 3) Direct Credit 4) Credit Card



Log Out

BACK PANEL ADMINISTRATION

Creating new cards

Last digit card \*

Card Type \*  
Select

Name Person assigned to card \*


Card expiration date \*

RETURN TO DRAFTS

SAVE

**New Registration Cards**

The Credit Card registration module lets you add a new CC on record. To add a new Credit Card you must fill-in some required fields. If the expiration date on a card is close to end, the system will send an alert by mail.

Log Out

BACK PANEL ADMINISTRATION

Creating new business unit

Number Business Unit \*

Name Business Unit \*


Previous name Business Unit \*

RETURN TO DRAFTS

SAVE

**Business Registration Unit**

The Business Unit Registration module lets you add different business units that may need contract management.

Log Out

BACK PANEL ADMINISTRATION

Creating new purchase order

Budgetary code \*

Previous value

Date last purchase order \*

Last invoice number \*

FILE


Upload a file

CANCEL

SAVE

Registration Purchase Order

The purchase order registration module allows you to add all possible purchase orders have been generated for a contract. This purchase order control then serve to better order when creating a new contract..

Log Out

BACK PANEL ADMINISTRATION

Creating a New Contract

Name of Contract \*

Contract Number

Type of contract \*

Select

Contract Status \*

Select

Approver \*

Select

Administrator \*

Select

Manufacturing Supplier \*

Select

Distributor Supplier \*

Select

Product \*

Select

Number Licenses

Budget Code

Only numbers allowed entry

Select

¿Save as Draft?

OFF

ON

Recomendacion #2


Debe seleccionar un proveedor

Manufacturero primero

Next

Registration Contracts

The contract registration module is the main module of the system, in this module you can create a new contract which has been established with any supplier, specify the category, area of the contract, indicate the start dates, completion and payment dates of the contract. By indicating these elements the system will have the ease of generating mail alerts to indicate when the next payment date or a contract is near its end date.


Log Out

**BACK PANEL ADMINISTRATION**
Save as Draft?
OFF
ON

### Creating a New Contract

Name of Contract \*
Contract Number

Type of contract \*  
Select

Contract Status \*  
Select
Service Contract Number

Approver \*  
Select
Administrator \*  
Select

Manufacturing Supplier \*  
Select
Distributor Supplier \*  
Select

Product \*  
Select

Number Licenses  
Only numbers allowed entry
Budget Code  
Select

### Registration Contracts

The contract registration module is the main module of the system, in this module you can create a new contract which has been established with any supplier, specify the category, area of the contract, indicate the start dates, completion and payment dates of the contract. By indicating these elements the system will have the ease of generating mail alerts to indicate when the next payment date or a contract is near its end date.

### Contract Payments

Payment Method \*  
Select
Card Type  
Select

Payment type \*  
Select
Total cost (\$) \*

Cost for each payment (\$) \*
Warranty

Warranty expiration date
Renewal Payment Date \*

### Contract Area

Business Unit Name \*  
Select
Contract Area \*  
Select

### Term of Agreement

Start Date \*
Finish Date \*

Warranty expiration date  Renewal Payment Date \*

**Contract Area**

Business Unit Name \*  Contract Area \*

**Term of Agreement**

Start Date \*  Finish Date \*

**Others**


Description Contract

A brief description of the contract

Set time alerts \*

**FILE** Upload a file

**SAVE AS DRAFT** **SAVE**


Log Out

### Contracts Registered in the system

Search:

Contract Number	Name of Contract	Type of contract	Cost for each payment	Payment type	Renewal Payment Date	State	Edition
	Kaltura	License	236790.000	Annual	3/31/2017	Active	<a href="#">Edit</a>   <a href="#">Delete</a>
7101-3889-8419-	Google Apps	License	25.000	Monthly	8/3/2016	Active	<a href="#">Edit</a>   <a href="#">Delete</a>
746822605337	Amazon Web Services	Maintenance	2400.000	Monthly	8/3/2016	Active	<a href="#">Edit</a>   <a href="#">Delete</a>

Showing 1 to 3 of 3 entries

Previous 1 Next

**BACK PANEL ADMINISTRATION**

