**Introducción.**

**Capítulo 1: Marco de referencia.**

* 1. **Sobre la empresa.**

**Historia**

El proyecto comenzó en 1996 con la creación de INSSEC y su participación en Corporación Industrial Egaña y el Grupo CIE Automotive como tal surgió en 2002 como resultado de la fusión entre grupo Egaña y el Grupo Aforasa.

**Localización.**

CIE Automotive diseña y fabrica globalmente componentes y subconjuntos en 98 centros productivos.

Nuestras plantas (8 de ellas multitecnológicas) están ubicadas en 90 localizaciones diferentes y se encuentran repartidas por 17 países en 4 continentes: Norteamérica (10 México y 5 EEUU), Sudamérica (12 Brasil), Europa (21 España, 2 Francia, 2 Portugal, 1 Reino Unido, 3 Alemania, 1 Italia, 6 Republica Checa, 2 Rumania, 1 Lituania, 1 Rusia, 1 Eslovaquia), Asia (18 India y 3 China) y África (1 Marruecos).

La multilocalización es nuestra propia versión del habitual proceso de internacionalización, basada en la gestión descentralizada y liderada por equipos locales.



* 1. **Filosofía Empresa.**
     1. **Misión.**

CIE Automotive es un proveedor de componentes y subconjuntos para el mercado global automotriz, con una actuación basada en la utilización de tecnologías complementarias y diversos procesos asociados.

Con una visión integrada de toda la cadena de valor.

Crecemos de forma sostenida y rentable para posicionarnos como socio de referencia a través de la satisfacción de nuestros clientes con soluciones integrales, innovadoras y competitivas de alto valor agregado.

Buscamos la excelencia sobre la base de los siguientes compromisos:

* La mejora continua de los procesos y su gestión eficiente.
* La promoción de la participación, la implicación y el trabajo en equipo en un entorno agradable y seguro.
* La transparencia y la integridad en todas nuestras actuaciones.
* El respeto y mejora del medio ambiente.
  + 1. **Visión.**

Aspiramos a convertirnos en el grupo industrial de referencia especialista en gestión de procesos de alto valor añadido.

Buscamos convertirnos en el paradigma de empresa socialmente responsable por nuestro compromiso con:

* Las personas y sus derechos fundamentales.
* El medio ambiente, fomentando iniciativas que promuevan una mayor responsabilidad ambiental.
* La creación de valor.
* La colaboración con grupos de interés.
* La excelencia en la gestión.

Nos proponemos ser:

* Referencia dentro de la cadena de valor por calidad, tecnología y servicios.
* Referencia en eco innovación y eco diseño.
  + 1. **Valores.**

En CIE Automotive damos importancia a las personas:

* Respetando sus derechos fundamentales.
* Proveyéndoles de unas condiciones laborales justas.
* Fomentado su capacidad de iniciativa, creatividad e innovación, la participación y el trabajo en equipo, su capacidad de conseguir objetivos y añadir valor, la actitud positiva al cambio y a la mejora continua.

En CIE Automotive damos importancia al medio ambiente:

* Manteniendo un enfoque preventivo.
* Trabajando para minimizar cualquier impacto negativo.

En CIE Automotive damos importancia a la transparencia en la gestión:

* Promoviendo la responsabilidad, la integridad y el compromiso con el trabajo bien hecho.
* Haciendo públicos de forma clara todos los datos relevantes de nuestra actividad para que éstos sean conocidos y comprendidos.

En CIE Automotive damos importancia a los grupos de interés:

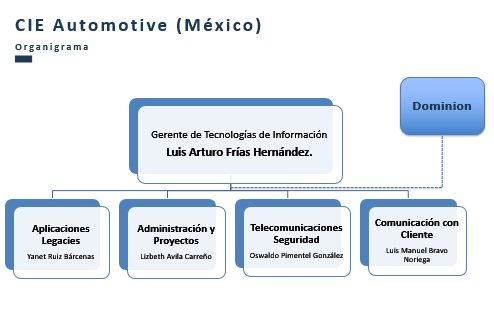
* Promoviendo unas relaciones honestas.
* Respetando sus derechos.

En CIE Automotive damos importancia a la legalidad:

* Respetando la normativa nacional e internacional.

HONESTIDAD, EQUIDAD E INTEGRIDAD SON LA BASE DE TODOS NUESTROS VALORES.

* + 1. **Organigrama.**

****

* 1. **Descripción del departamento donde se desarrollan las actividades.**

**Nombre del departamento:** Tecnologías de la Información.

**Nombre del proyecto:** “Administración de Licenciamiento CIE Automotive México”.

**Asesor externo/ Encargado de proyecto:** Lic. Lizbeth Avila Carreño.

* + 1. **Objetivos Generales del Proyecto.**

Desarrollar un sistema de administración de licencias de software, mediante el uso de herramientas de desarrollo de software para mantener un mejor control de las licencias de software utilizadas en los proyectos de las diferentes áreas de producción que conforman el grupo CIE Automotive México.

* + 1. **Objetivos Específicos del proyecto.**

1. Garantizar que todo el software utilizado en los proyectos de la corporación sea de procedencia legitima.
2. Dar apoyo al área administrativa del departamento de TI, mediante una herramienta que almacene toda la información relacionada a licenciamiento.
3. Asegurar el cumplimiento de pago a tiempo de licencias.
4. Promover el uso de software legal.
5. Tener una clasificación de proveedores con una correcta definición de que software proporciona cada uno de ellos.
   1. **Problemática a resolver.**

Grupo CIE Automotive México, es una empresa socialmente responsable y apegada a la normativa nacional e Internacional es por ello que entiende que el uso de software pirata no sólo es ilegal, sino que también representa un retroceso en la competitividad, productividad y mejora de la industria, así como entiende que el adquirir, descargar e instalar software de procedencia no licita puede resultar en riesgos importantes, tanto legales como de seguridad.

El proyecto “Administración de licenciamiento CIE Automotive” consiste en el desarrollo de un sistema que permita al departamento de Tecnologías de la Información una administración más óptima de las licencias con que actualmente cuenta la empresa y aquellas que pueden ser adquiridas para proyectos futuros que permiten cumplir con las actividades de las diferentes áreas tanto productivas como administrativas de la empresa siempre apegados a la normativa de uso.

La necesidad de este proyecto surge de la importancia de poder acceder a dichas licencias en el momento que se requieran de una forma más organizada. Siendo esto, que cualquier usuario con acceso al sistema pueda ver qué equipo de cómputo tiene asignada cuál licencia, es decir ver las licencias ya asignadas, insertar nuevas y asignarlas y realizar si así lo requiere algún cambio, así mismo, poder dar de baja temporal licencias que de momento no están siendo utilizadas pero de las cuales la empresa ya es propietaria, así como conocer los beneficios que otorga cada licencia, esto porque algunas modalidades contienen servicios extra mejor conocido como un Kit, otras simplemente son por el uso de un determinado software, de aquí la importancia de conocer a detalle cada licencia y sus características. Una de las funciones más importantes de dicho sistema es permitirle al encargado del área de TI, saber las fechas de vencimiento de las licencias o la fecha de pago por el mantenimiento, esto dependiendo de la modalidad que maneje la licencia.

Por lo tanto, es importante conocer todo el software que es utilizado, así como lo son los de distribución libre que no generan costos y aquellos que dan lugar a este proyecto y todos los detalles del contrato bajo el cual se adquirieron cada uno de ellos.

El diseño del software está pensado para ser utilizado por una cantidad de usuarios reducida, que son aquellos pertenecientes al departamento de TI y que tienen en sus actividades la administración de licencias de software. Por lo tanto, se pretende que aquellos usuarios que lo utilicen sean únicamente personas que entiendan lo correspondiente a licenciamiento.

* 1. **Justificación.**

El software pirata es un problema que afecta a más de la mitad de las compañías en México. De acuerdo con la “Encuesta global de Software” realizada por la BSA [1] se encontró que el porcentaje de empresas en México que trabajan con software pirata [1] se encuentra entre el 54% para el 2013 y 52% para el 2015. Siendo esta disminución producto del reconocimiento de directivos de la relación que existe entre el uso de software sin licencia y las amenazas de seguridad. Este proyecto pretende desarrollar un sistema de gestión de licencias de software que permita al grupo CIE, prevenir riesgos fiscales, así como cumplir con uno de sus principales objetivos, ser una empresa socialmente responsable que promueve el no uso ilegal de software. Esto beneficia a todo el conjunto de empresas que conforman el grupo CIE Automotive México.

[1]Business Software Alliance |BDO México

[2] Software pirata es un término utilizado para describir la reproducción, distribución, comercialización y el uso de software de cualquier forma no autorizada, lo cual está prohibido en Mexico por la Ley Federal de Derechos de Autor (LFDA) y el Código Penal Federal (CPF)

* + 1. **Alcances y limitaciones.**

**Capítulo 2. Marco Teórico.**

1. **Visual Studio.**

**Microsoft Visual Studio** es un entorno de desarrollo integrado (IDE por sus siglas) para sistemas operativos Windows.

Soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic, .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP .NET MVC, Django, etc., a lo cual hay sumarle las nuevas capacidades online bajo Windows Azure.

Un IDE es un entorno de desarrollo integrado en inglés Integrated Development Enviroment, es una aplicación informática que proporciona servicios integrales para facilitarle al programador el desarrollo de software.

Un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador.

1. **ASP .NET**

ASP .NET es una plataforma web que proporciona los servicios necesarios para compilar aplicaciones web empresariales basadas en servidor ASP.NET está compilado en .NET Framework, por lo que todas las características de .NET Framework están disponibles en las aplicaciones ASP.NET. Las aplicaciones se pueden escribir en cualquier lenguaje que sea compatible con Common Language Runtime (CLR), incluido Microsoft Visual Basic, C# (C Sharp), JScript .NET.

El marco de trabajo que se utiliza en el desarrollo del proyecto se denomina **ASP .NET MVC,** este no sustituye al modelo de formularios Web Forms, proporciona una alternativa más para crear aplicaciones web.

ASP .NET MVC en un marco de presentación de poca complejidad y fácil de comprobar que (como las aplicaciones basadas en formularios Web Forms) se integra con las características de ASP .NET existentes, tales como paginas maestras y la autenticación basada en pertenencia.

1. **MVC**

EL MVC Modelo-Vista-Controlador es un patrón de arquitectura de software que, utilizando 3 componentes (Vistas, Models y Controladores).

El patrón de diseño de modelo-vista-controlador (MVC) especifica que una aplicación consta de un modelo de datos, de información de presentación y de información de control. El patrón requiere que cada uno de estos elementos esté separado en distintos objetos.

A continuación, se describen los elementos del patrón de arquitectura MVC:

--Brenda del futuro, aquí debes poner una imagen representativa de MVC.

**Modelo:** Los objetos de modelo son las partes de la aplicación que implementan la lógica de dominio de datos de la aplicación. contiene la estructura de los datos, las clases y entidades, en definitiva, el modelo de datos.

Así mismo, es la que proporciona a la vista y al controlador la información tanto a nivel de estructura como a nivel de datos.

Es en esta capa donde se produce la comunicación con la base de datos.

**Vistas:** es el módulo de la aplicación que interactúa con el usuario. Recibe los datos, ejecuta eventos y muestra la información que recibe el controlador. En programación web forma parte del Front-End y normalmente está programado en HTML, JavaScript y CSS.

**Controladores:** Son los componentes que controlan la interacción del usuario, son el enlace entre la vista y el modelo.

Escucha los sucesos desencadenados por la vista (u otro origen externo) y ejecuta la reacción apropiada a estos sucesos. En la mayoría de los casos, la reacción es llamar a un método del modelo. Puesto que la vista y el modelo están conectador a través de un mecanismo de notificación, el resultado de esta acción se reflejará automáticamente en la vista.

**Características del marco de ASP .NET MVC**

--Brenda del futuro, resume las características

* Separación de tareas de aplicación (lógica de entrada, lógica de negocios y lógica de la interfaz de usuario), facilidad para pruebas y desarrollo basado en pruebas (TDD).
* Un marco extensible y conectable. Los componentes del marco de ASP.NET MVC están diseñados para que se puedan reemplazar o personalizar con facilidad.
* Amplia compatibilidad para el enrutamiento de ASP.NET, un eficaz componente de asignación de direcciones URL que le permite compilar aplicaciones que tienen direcciones URL comprensibles y que admiten búsquedas. Las direcciones URL no tienen que incluir las extensiones de los nombres de archivo y están diseñadas para admitir patrones de nombres de direcciones URL que funcionan bien para la optimización del motor de búsqueda (SEO).
* Compatibilidad para usar el marcado en archivos de marcado de páginas de ASP.NET existentes (archivos .aspx), de controles de usuario (archivos .ascx) y de páginas maestras (archivos .master) como plantillas de vista.
* Compatibilidad con las características de ASP.NET existentes. ASP.NET MVC le permite usar características, tales como la autenticación de formularios y la autenticación de Windows, la autorización para URL, la pertenencia y los roles, el caching de resultados y datos, la administración de estados de sesión y perfil, el seguimiento de estado, el sistema de configuración y la arquitectura de proveedor.

**Ventajas de una aplicación web basada en MVC.**

* Facilita la administración de la complejidad, al dividir una aplicación en el modelo, la vista y el controlador.
* No usa el estado de vista ni formularios basados en servidor. Esto hace que el marco MVC sea ideal para los desarrolladores que deseen un control completo sobre el comportamiento de una aplicación.
* Proporciona una mayor compatibilidad con el desarrollo basado en pruebas (TDD).
* Funciona bien para las aplicaciones web en las que trabajan equipos grandes de desarrolladores y para los diseñadores web que necesitan un alto grado de control sobre el comportamiento de la aplicación.

Seguir el patrón MVC, permite una migración de motor de base de datos mucho más simple, comparado con el acceso a la base de datos desde las tres capas (Modelo, Vista, Controlador) esto, ya que implica realizar cambios únicamente en la capa de acceso a datos que es la capa del Modelo y no en cada capa como sería en el caso contrario.

De esta forma, en la capa de Modelo se suele utilizar los ORM, que simplifican la migración de un motor de base de datos a otro.

1. **ORM**

Un Mapeo Objeto-Relación (ORM, del inglés Object Relational Mapping) permite convertir los datos de los objetos en un formato correcto para poder guardar la información en una base de datos (mapeo) creándose una base de datos virtual donde los datos que se encuentran en la aplicación queden vinculados a la base de datos (persistencia).

En una aplicación que se conecta a una base de datos, el mapeo es la transformación de la información que se recibe de una base de datos, principalmente en tablas, en objetos de la aplicación y viceversa.

Utilizando un ORM este mapeo será automático, independiente de la base de datos que se utilice en el momento, lo que se traduce en la posibilidad de cambiar de motor de base de datos según las necesidades.

1. **Entity Framework (.NET)**
2. **C# (C Sharp).**

**C#** es un lenguaje de programación orientado a objetos, desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

**Se** deriva de C y C++, es moderno, simple y enteramente orientado a objetos, simplifica y moderniza a C++ en las áreas de clases, namespaces, sobrecarga de métodos y manejo de excepciones.

Se eliminó la complejidad de C++ para hacerlo más fácil de utilizar y menos propenso a errores.

C# permite a los desarrolladores crear una gran variedad de aplicaciones seguras y sólidas que se ejecutan en .NET Framework .NET. C# se puede utilizar para crear aplicaciones cliente de Windows, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos y muchas más.

1. **HTML.**
2. **Razor**
3. **Bootstrap**

Bootstrap es un conjunto de conceptos, prácticas y criterios (framework) desarrollado por Mark Otto y Jacoob Thornton dentro de Twitter con la intención de estandarizar el conjunto de herramientas que utilizaban todos los involucrados en el desarrollo de Front-End.

Bootstrap es uno de los principales frameworks de desarrollo de entornos web “responsive” (es decir, que se adaptan automáticamente al tamaño de la pantalla de cada usuario).

Esencialmente Bootstrap está formado por una colección de hojas de estilo y funciones JavaScript auxiliares que permiten construir muy rápidamente un Front-End responsive.

Está pensado para trabajar bajo el concepto “mobile first”, es decir, diseñar pensando ante todo en los dispositivos móviles.

1. **CSS**
2. **SQL**

**Capítulo 3. Desarrollo del proyecto.**

**Capítulo 4. Competencias desarrolladas y/o aplicadas.**

**Referencias bibliográficas y virtuales.**

* **BSA (Business Software Alliance (BSA) / BDO México)**

Derecho de Autor BSA. (25 mayo 2016). Software ilegal, causa principal de los riesgos en ciberseguridad. Recuperado de: <https://goo.gl/LKZBnu>

* **IMPI (Instituto Mexicano de la propiedad Industrial)**

Instituto Mexicano de la Propiedad Industrial. (6 de febrero 2018). Recuperado de: <https://goo.gl/EG5UM6>

**Referencias del marco teorico.**

**ASP .NET**

<https://msdn.microsoft.com/es-es/library/4w3ex9c2(v=vs.100).aspx>

[**https://msdn.microsoft.com/es-es/library/dd381412(v=vs.108).aspx**](https://msdn.microsoft.com/es-es/library/dd381412(v=vs.108).aspx)

**MVC.**

<https://www.ibm.com/support/knowledgecenter/es/SSZLC2_8.0.0/com.ibm.commerce.developer.doc/concepts/csdmvcdespat.htm>

<https://programarfacil.com/podcast/19-patron-modelo-vista-controlador/>

<https://programarfacil.com/blog/que-es-un-orm/>

**Entity Framework**

**C# (C- Sharp)**

[**https://docs.microsoft.com/es-es/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework**](https://docs.microsoft.com/es-es/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework)

**Bootstrap**