

**S.E.S.T.N.M.**

**T.N.M.**

**S.E.P.**

# INSTITUTO TECNOLÓGICO DE TOLUCA

# INGENIERÍA EN SISTEMAS COMPUTACIONALES

# INFORME TÉCNICO DE RESIDENCIA PROFESIONAL

**“SISTEMA DE PLANEACIÓN INSTITUCIONAL DE CECYTEM” (SPIC).**

**P R E S E N T A:**

**Adolfo Emmanuel Arriaga Vargas**

**NO. CONTROL:**

**10280564**

**ASESOR INTERNO:**

**M. en C. Mauro Sánchez Sánchez**

**ASESOR EXTERNO:**

**L.I.A Dante Salvador Ortega Aguilar**

## 

#### METEPEC, ESTADO DE MÉXICO, DICIEMBRE DEL 2016.

### 

### Indice

[Indice - 2 -](#_Toc735409006)

[1.- Antecedentes 3](#_Toc620154676)

[2.- Planteamiento del Problema 6](#_Toc1991824186)

[3.- Objetivos 7](#_Toc2020279914)

[3.1.- Objetivo General 7](#_Toc1580462842)

[3.2.-Objetivos Específicos 7](#_Toc694333784)

[4.- Justificación 8](#_Toc1412209716)

[5.- Marco Teórico 9](#_Toc1188950267)

[5.1.- Ingeniería del Software 9](#_Toc1672551351)

[5.2.- Metodología de Desarrollo 9](#_Toc1502859136)

[5.2.1.- Metodologías del Software 9](#_Toc1289849705)

[5.2.2.- Metodología de Prototipado y el modelo de proceso evolutivo 10](#_Toc2004738213)

[5.3.- Modelado de sistemas de información 12](#_Toc758054599)

[5.3.1.-Diagrama de casos de uso 13](#_Toc1205650079)

[5.3.2.- Diagrama de Clases 14](#_Toc1110509431)

[5.3.3.- Diagrama de Actividades 19](#_Toc826320497)

[5.4.- Sistemas de Información 19](#_Toc1196394831)

[5.5.- Arquitectura de los Sistemas Web 21](#_Toc1593891356)

[6.- Desarrollo 24](#_Toc1233016517)

[6.1.- Análisis del Sistema 25](#_Toc1967981186)

[6.1.- Diseño de aplicación 28](#_Toc858640801)

[7.- Resultados 29](#_Toc850496660)

[8.- Conclusiones 30](#_Toc451674344)

[9.- Referencias 31](#_Toc1075064557)

[10. - Anexos 32](#_Toc1203031644)

[10.1 Anexo A 32](#_Toc1885913220)

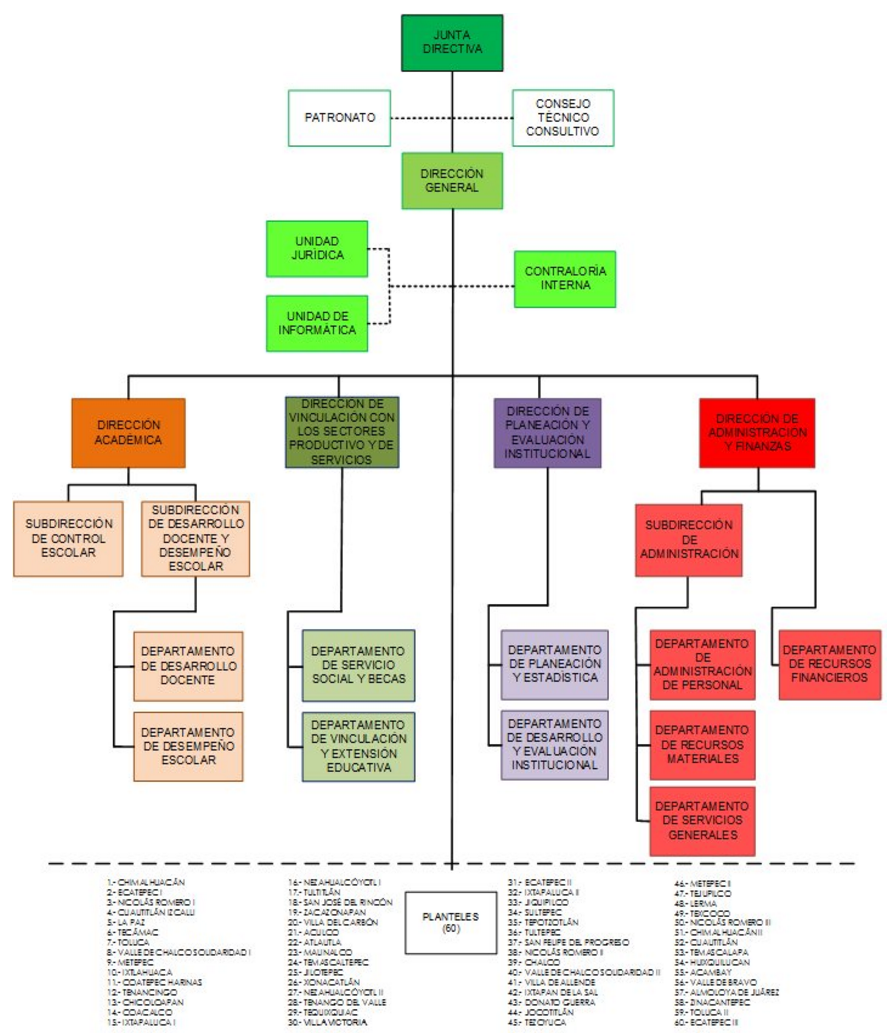
# 1.- Antecedentes

En la actualidad, el software tiene un papel dual. Es un **producto** y al mismo tiempo es el **vehículo** para entregar un producto. En su forma de **producto**, brinda el potencial incorporado en el hardware de cómputo. Ya sea que resida en un teléfono móvil u opere en el interior de una computadora central, el software es un transformador de información, produce, administra, adquiere, modifica, despliega o transmite información que puede ser tan simple como un solo bit o tan compleja como una presentación con multimedios generada a partir de datos obtenidos de decenas de fuentes independientes. Como **vehículo** utilizado para distribuir el producto, el software actúa como la base para el control de la computadora (sistemas operativos), para la comunicación de información (redes) y para la creación y control de otros programas (herramientas y ambientes de software).

El software distribuye el producto más importante de nuestro tiempo: **información**. Transforma los datos personales de modo que puedan ser más útiles en un contexto local, administra la información de negocios para mejorar la competitividad, provee una vía para las redes mundiales de información (la Internet) y brinda los medios para obtener información en todas sus formas.

En la actualidad, la enorme industria del software se ha convertido en un factor dominante en las economías del mundo industrializado.

El Colegio de Estudios Científicos y Tecnológicos del Estado de México es una institución estatal de educación media superior derivada de la Secretaría de Educación Publica del Estado de México. El colegio se compone de 60 planteles en diferentes municipios del estado, y a su vez oficinas centrales que se encargan de gestionar los procesos correspondientes de CECYTEM. En la **Figura 1** se muestra el organigrama del colegio.



**Figura 1. Organigrama de CECYTEM.**

Desde la creación del Colegio de Estudios Científicos y Tecnológicos del Estado de México, la información a cambiado de forma radical en cuanto a la cantidad que se genera, como en la forma de manejar esta información. Poco a poco el Colegio se ha ido adaptando a los cambios tecnológicos tanto en la infraestructura (redes de computadora, telefonía, sistemas de información) como en el uso de estas tecnologías.

Al paso de los años se han implementado sistemas de información para remplazar procesos que normalmente se hacían de forma manual. La forma de administrar, producir, desplegar información ha cambiado en algunas áreas de la organización tal es el caso de los departamentos Académico y Recursos Materiales en los que el personal administra la información respectivamente, en base a las aplicaciones software con las que cuenta, mismas que se han desarrollado a lo largo de los últimos cuatro años. Se ha logrado Implementar sistemas en los que destaca el Sistema de Control Escolar (DEO), así como el Sistema de Inventario para Bienes Informáticos (SIBIEN).

El desempeño, la facilidad y la confiabilidad de estos sistemas de información han permitido operar de manera rápida y sencilla en las áreas mencionadas, generando zonas de oportunidad para sistematizar otros procesos en mas áreas de la organización como: el departamento de Planeación y Evaluación Institucional, Administración y Finanzas, Contraloría etc.

# 2.- Planteamiento del Problema

El departamento de Planeación y Evaluación Institucional es el encargado de controlar toda la información de planes y programas de cada una de la unidades administrativas y planteles que conforman el colegio, así como la evaluación de estos rubros, en base a esto se determinan estrategias, políticas y procedimientos para la realización de las actividades de toda la organización. Adquiriendo una enorme cantidad de información en cada uno de los procesos de dicho departamento.

Parte de la información que se genera es la de las fichas técnicas de planteles, las cuales recopilan información general, académica y de infraestructura de estos planteles, las cuales se ocupan para planificar el desarrollo de infraestructura, así como saber información académica importante (Personal, matricula, docentes).

La necesidad de mantener la información disponible, confidencial e integra de las fichas técnicas de los 60 Planteles es evidente, para las funciones inherentes del área. Hasta la fecha toda esa información se mantiene en físico (papel), haciendo que el control no sea el óptimo, no este disponible de manera rápida y deficiente cuando se requiere, ademas de que el contenido de la ficha va cambiando semestre por semestre, esta información se resguarda, haciendo ineficiente la administración, ocupando mas espacio y desperdiciando mucho papel. Aunque existen procedimientos administrativos para el manejo de la información existen conflictos en la implementación, por falta de capacitación a los servidores públicos.

# 3.- Objetivos

## 3.1.- Objetivo General

Desarrollo e Implementación de una aplicación web para el departamento de Planeación y Evaluación del CECYTEM (Sistema de Planeación Institucional de CECYTEM “SPIC”) para gestionar, actualizar y desplegar las fichas técnicas de cada plantel que integra el organismo.

## 3.2.-Objetivos Específicos

Los Objetivos se basan en las fases de desarrollo del Sistema “SPIC” por lo que se han determinado los siguientes objetivos específicos:

1. Diseñar una base de datos correspondiente al Sistema de Información “SPIC” que se adapte a la estructura de la base de datos del Sistema de control Escolar “DEO”
2. Mantener la Integridad de los datos de acuerdo a la información contenida en el Sistema “DEO”.
3. Desarrollo e Implementación del Sistema “SPIC” de acuerdo a los estándares de desarrollo del CECYTEM.
4. Desarrollo del módulo de creación, actualización y visualización de las fichas Técnicas.
5. Desarrollo del módulo de etapas de infraestructura para los planteles del CECYTEM.
6. Desarrollo del módulo de acceso al sistemas “SPIC”.

# 4.- Justificación

Hoy en día la tecnología ocupa un papel muy importante en la manera de llevar toda la información de las organizaciones públicas y privada, haciendo de los sistemas de información una parte vital en los procesos y actividades de todos los departamentos que las integran, para procesar, operar y mostrar información en el momento que se requiera.

Los sistemas de información cubren las necesidades mencionadas facilitando a los procesos críticos de las organizaciones, la seguridad y fiabilidad al ser ejecutados.

Un sistema de información que lleve todo el flujo y procesamiento de la información en una organización es esencial hoy en día. Por eso se presenta la necesidad de un sistema que maneje, procese y despliegue fichas técnicas con información académica y de infraestructura de cada plantel que conforma el CECYTEM, que a su vez el departamento de Planeación y Evaluación Institucional genera.

El sistema “SPIC” (Sistema de Planeación Institucional de CECYTEM) se encarga del manejo de las Información relacionada a las etapas de Infraestructura de cada plantel, de manera ordenada y estructurada.

Esto trae diferentes beneficios al departamento de Planeación, ya que toda la información la tienen contenida en un solo lugar, de manera digital, de acceso restringido y fácil manejo. El sistema reduce tiempos al procesar la información, elimina el uso innecesario de papel y mantiene la confidencialidad y disponibilidad de la información.

5.- Marco Teórico

## 5.1.- Ingeniería del Software

Los sistemas de información comprenden un conjunto de tecnologías, procesos de negocio y de software disponibles para las personas y/o empresas. Un sistema de Información no sera funcional y optimo si no se llevara un análisis, diseño y desarrollo adecuada sin planificar adecuadamente una planeación es por esto que la ingeniería del software ayuda a optimizar tiempos y recursos de una manera adecuada estas fases. Sommerville (2005) afirma “La ingeniería de software es la actividad de especificar, diseñar, implementar, validar, utilizar y mantener los sistemas de información ”.

Los ingenieros de sistemas no solo tratar con el software sino también con hardware y las interacciones del sistema con los usuarios y su entorno. Deben de pensar en los servicios que el sistema proporciona, las restricciones y las formas en las que el sistema cumple su propósito.

La creación del software es un proceso intrínsecamente creativo y la ingeniaría del software trata de sistematizar este proceso con el fin de acortar el riesgo del fracaso al perseguir un objetivo en especifico.

## 5.2.- Metodología de Desarrollo

### 5.2.1.- Metodologías del Software

Un método para el desarrollo de software es una representación abstracta de un proceso. Cada método representa un proceso desde una perspectiva particular para proporcionar información sobre el proceso, se pueden pensar como marcos de trabajo del proceso y que pueden ser adaptados para crear procesos mas específicos.

Al seguir una metodología se hace uso de diversas herramientas, técnicas, métodos y modelos para el desarrollo. Actualmente existen mucha variedad en metodologías de desarrollo en la que se pueden basar dependiendo del enfoque y las necesidades de los proyectos.

Para efectos de este desarrollo se estudia la metodología de prototipado.

La metodología en la que se ha basado el desarrollo del sistema es la denominada **prototipado**, en base a los requerimientos establecidos por el departamento de Planeación y Evaluación Institucional y basándose en el tiempo de desarrollo y entrega, se ha determinado esta metodología.

**Metodología de Prototipado.**

La Metodología de Prototipado permite que todo o parte del sistema se construya rápidamente y poder así comprender con facilidad los aspectos que aseguren que el desarrollador, el usuario y el cliente estén de acuerdo en lo que el sistema necesita , así como también la solución que se propone para dicha necesidad y de esa manera poder minimizar el riesgo y la incertidumbre en el desarrollo. La Metodología se basa en diseños del sistema para que sean analizados y prescindir de ellos a medida que se acoplen a las nuevas especificaciones, de esa manera de determina con mayor facilidad el alcance del sistema.

El desarrollo se basa en entregables (Prototipos), de esta manera se puede ver la funcionalidad básica de un sistema, sin necesidad de incluir toda la lógica o características del software terminado, los prototipos permiten al cliente evaluar en cada etapa el producto, e interactuar con los diseñadores y desarrolladores para saber si se esta cumpliendo con las expectativas y las funcionalidades establecidas. Los prototipos no contienen la funcionalidad total del sistema pero ayuda a estructurar la idea principal del mismo.

### 5.2.2.- Metodología de Prototipado y el modelo de proceso evolutivo

El software, así como los sistemas complejos, evolucionan con el tiempo. Es posible que los requerimientos y especificaciones del negocio y del producto cambien conforme avanza el desarrollo, los plazos apretados del mercado hacen que sea imposible la terminación de un software perfecto, pero debe lanzarse una versión limitada a fin de aliviar la presión de la competencia o del negocio, se contemplan un conjunto de requerimientos mínimos que liberen esa presión, para estas situaciones se necesita un modelo de proceso diseñado explícitamente con el objeto que se adapte a un producto que evoluciona con el tiempo, dichos modelos evolutivos son iterativos y se caracterizan por la manera en la que permiten desarrollar versiones cada vez mas complejas de software.

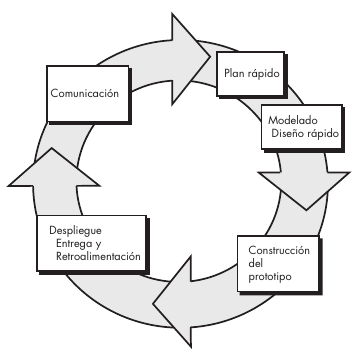
Es frecuente que el cliente defina un conjunto de objetivos generales para el software, sin embargo no tiene contemplado los requerimientos detallados para las funciones y características que el sistema deberá tener, en otros casos que el desarrollador no este seguro la eficiencia de su algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debe de adoptar la interacción entre el humano y la maquina. En estas situaciones, y muchas otras, el paradigma de hacer prototipos ofrece mejor enfoque (Pressman Roger S, 2010).

El paradigma de hacer prototipos ayuda a que los participantes del desarrollo mejoren la comprensión de lo que hay que elaborar cuando los requerimientos no son claros.

**Ciclo de vida del desarrollo basado en prototipos**

El paradigma de hacer prototipos (**Figura X**), comienza con la comunicación, se realizan una serie de reuniones con los participantes del proyecto para determinar los objetivos generales del software, se identifica los requerimientos reales y se detecta las áreas en las que es imprescindible una mayor atención. Se planea una iteración para hacer prototipos, y se lleva acabo un diseño rápido. Se centra en la representación de aquellos aspectos del software que serán visibles para los usuarios finales (por ejemplo, disposición de interfaces o formatos de pantalla de salida). El diseño rápido lleva a la construcción de un prototipo. Este se entrega y es evaluado por los participantes, que dan retroalimentación para mejorar los requerimientos. Las iteraciones ocurren a medida que el prototipo es afinado para satisfaces las necesidades de distintos participantes, y al mismo tiempo permite comprender mejor lo que se necesita hacer.

Lo ideal de un prototipo es que sirva como mecanismo para identificar los requerimientos del software. Se puede utilizar fragmentos de programas existentes o herramientas que permitan generar rápidamente programas que funcionen.



**Fig. X -Ciclo de vida de Prototipos.**

La pregunta que se formula al emplear un modelo de este tipo es:

**¿Que hacer con el prototipo cuando ya sirvió para el propósito descrito?**

En la mayoría de los proyectos es raro que el primer sistema hecho sea utilizable. Tal vez sea muy lento, muy grande, difícil de usar o todo a la vez. No hay alternativa que comenzar de nuevo, con mas inteligencia, y construir una versión rediseñada en la que se resuelvan los problemas.

## 5.3.- Modelado de sistemas de información

El análisis de los requerimientos da como resultado la especificación de las características operativas del software, indica la interfaz de éste y otros elementos del sistema, y establece las restricciones que limitan al software. El análisis de los requerimientos permite al profesional (sin importar si se llama ingeniero de software, analista o modelista) construir sobre los requerimientos básicos establecidos durante las tareas de concepción, indagación y negociación, que son

parte de la ingeniería de los requerimientos.

La acción de modelar los requerimientos da como resultado uno o más de los siguientes tipos de modelo:

1. Modelos basados en el escenario de los requerimientos desde el punto de vista de distintos “actores” del sistema.
2. Modelos de datos, que ilustran el dominio de información del problema.
3. Modelos orientados a clases, que representan clases orientadas a objetos (atributos y operaciones) y la manera en la que las clases colaboran para cumplir con los requerimientos del sistema.
4. Modelos orientados al flujo, que representan los elementos funcionales del sistema y la manera como transforman los datos a medida que se avanza a través del sistema.
5. Modelos de comportamiento, que ilustran el modo en el que se comparte el software como consecuencia de “eventos” externos.

Estos modelos dan al diseñador del software la información que se traduce en diseños de arquitectura, interfaz y componentes. Por último, el modelo de requerimientos (y la especificación de requerimientos de software) brinda al desarrollador y al cliente los medios para evaluar la calidad una vez construido el software (Pressman Roger S, 2010).

**5.3.1.- Diagrama de casos de uso**

Los diagramas de casos de uso nos ayudan a determinar la funcionalidad y características del software desde la perspectiva del usuario. Proporciona una aproximación del alcance del proyecto.

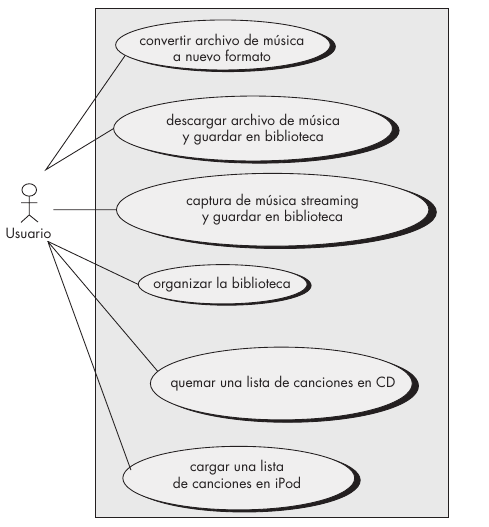
Un caso de uso describe la manera en la que un usuario interactúa con el sistema, definiendo los pasos requeridos para lograr una meta específica. Las variaciones en la secuencia de pasos describen varios escenarios. Un diagrama de este tipo es un panorama de todos los casos de uso y sus relaciones, este mismo proporciona un gran cuadro de la funcionalidad del sistema. En la **figura x** se muestra los componentes de este tipo de diagramas.



**Figura x. Componentes de diagrama de casos de uso**

* Los **actores** son entidades externas del sistema que guardan una relación con éste y que le demanda una determinada funcionalidad.
* Los **casos de uso** son las funciones o procesos que el sistema debe de ejecutar, estos casos pueden o no depender de otros actores u otros sistemas.
* Las **asociaciones** son los enlaces directos entre los actores y los casos de uso.
* Extensión o Inclusión son las relaciones de dependencia entre dos o mas casos de uso.
* Limite de sistema determina el alcance de las funciones y procesos que procesa el sistema.

En la **figura x** se muestra un ejemplo de caso de uso de un sistema de música, en el que se gestiona una biblioteca musical, teniendo diferentes funcionalidades.



**Figura x. Diagrama de casos de uso.**

**5.3.2.- Diagrama de Clases**

El diagrama de clases es un tipo de diagrama de estructura estática que describe a un sistema, mostrando las clases que la engloban, sus atributos, operaciones y relaciones entre los objetos. El modelos de clases sirve para visualizar las relaciones entre las clases que involucran al sistema, las cuales pueden ser asociativas, de herencia, de uso.

Un diagrama de clases esta compuesto por los siguientes elementos:

* [Clase](http://users.dcc.uchile.cl/~psalinas/uml/modelo.html" \l "clase): atributos, métodos y visibilidad.
* [Relaciones](http://users.dcc.uchile.cl/~psalinas/uml/modelo.html" \l "relacion): Herencia, Composición, Agregación, Asociación y Uso.

**Elementos de un diagrama de clase**

**Clase:** Es la unidad que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto). Como se muestra en la **figura x** una clase es representada por un rectángulo que posee tres divisiones.

**Figura x.- Objeto clase.**

**Atributos:** son variables de instancia que caracterizan a la clase. Los atributos o características de una Clase pueden ser de tres tipos:

* **public** (+): Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
* **private** (-): Indica que el atributo sólo será accesible desde dentro de la clase (sólo sus métodos lo pueden accesar).
* **protected** (#): Indica que el atributo no será accesible desde fuera de la clase, pero tiene acceso por métodos de la clase además de las subclases que se deriven (ver herencia).

**Métodos:** Los métodos u operaciones de una clase son la forma en como ésta interactúa con su entorno, éstos pueden tener las características:

* **public** (+): Indica que el método será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
* **private** (-): Indica que el método sólo será accesible desde dentro de la clase (sólo otros métodos de la clase lo pueden accesar).
* **protected** (#): Indica que el método no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de métodos de las subclases que se deriven (ver herencia).

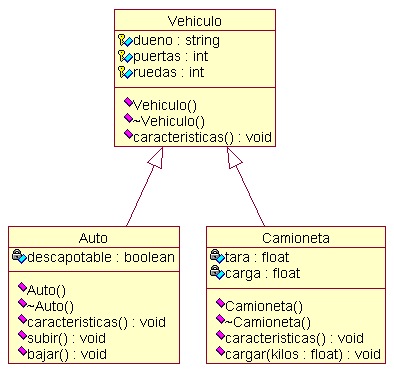
**Relaciones entre Clases**

Ahora ya definido el concepto de Clase, es necesario explicar como se pueden interrelacionar dos o más clases. Es necesario comprender el concepto de cardinalidad en las relaciones. La cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y éstas pueden ser:

* **uno o muchos**: 1..\* (1..n)
* **0 o muchos**: 0..\* (0..n)
* **número fijo**: m (m denota el número).

**Herencia (Especialización/Generalización)**:  IMG_264

Indica que una subclase hereda los métodos y atributos especificados por una Super Clase, por ende la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Super Clase (public y protected), en la **figura x** se muestra un ejemplo de diagrama de clase con herencia.

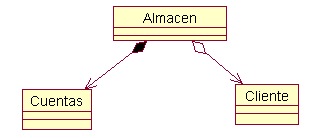


**Figura x.- Diagrama de clase con Herencia.**

**Agregación** IMG_266

Para modelar objetos complejos, no bastan los tipos de datos básicos que proveen los lenguajes: enteros, reales y secuencias de caracteres. Cuando se requiere componer objetos que son instancias de clases definidas por el desarrollador de la aplicación, tenemos dos posibilidades:

* **Por Valor**: Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido esta condicionado por el tiempo de vida del que lo incluye. Este tipo de relación es comúnmente llamada **Composición** (el Objeto base se construye a partir del objeto incluido, es decir, es "parte/todo").
* **Por Referencia**: Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada **Agregación** (el objeto base utiliza al incluido para su funcionamiento).

En la figura x se muestra una ejemplo asociativo.

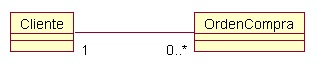
**Figura x.- Diagrama de Asociación.**

En donde se observa:

* Un Almacén posee Clientes y Cuentas (los rombos van en el objeto que posee las referencias).
* Cuando se destruye el Objeto Almacén también son destruidos los objetos Cuenta asociados, en cambio no son afectados los objetos Cliente asociados.
* La composición (por Valor) se destaca por un rombo relleno.
* La agregación (por Referencia) se destaca por un rombo transparente.

**Asociación**:  IMG_268

La relación entre clases conocida como Asociación, permite asociar objetos que colaboran entre si. Cabe destacar que no es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.

En la **figura x**  se observa que un cliente puede tener asociadas muchas Ordenes de Compra, en cambio una orden de compra solo puede tener asociado un cliente.

**Figura x.- Asociación de Clases.**

**Dependencia o Instancia (uso)**:  IMG_270

Representa un tipo de relación muy particular, en la que una clase es instanciada (su instancia es dependiente de otro objeto/clase). Se denota por una flecha punteada. El uso más particular de este tipo de relación es para denotar la dependencia que tiene una clase de otra, como por ejemplo una aplicación gráfica que instancia una ventana (la creación del Objeto Ventana esta condicionado a la instancia proveniente desde el objeto Aplicación):

IMG_271En la **figura x** cabe destacar que el objeto creado (en este caso la Ventana gráfica) no se almacena dentro del objeto que lo crea (en este caso la Aplicación).

**Figura x.- Instancia de una clase.**

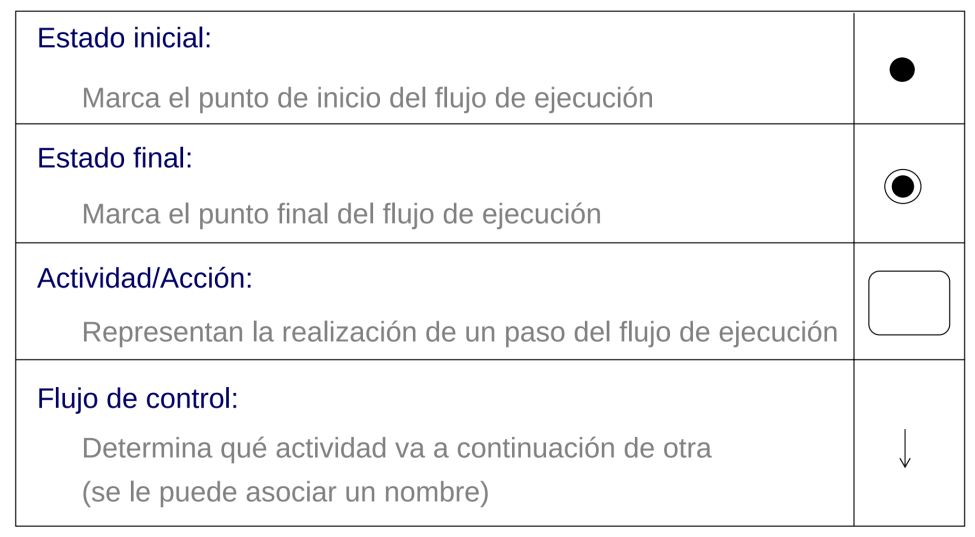
**5.3.3.- Diagrama de Actividades**

Un diagrama de actividades muestra un proceso de negocio o un proceso de software como un flujo de trabajo a través de una serie de acciones. Las personas, los componentes de software o los equipos pueden realizar estas acciones. Se puede usar un diagrama de actividades para describir procesos de varios tipos como:

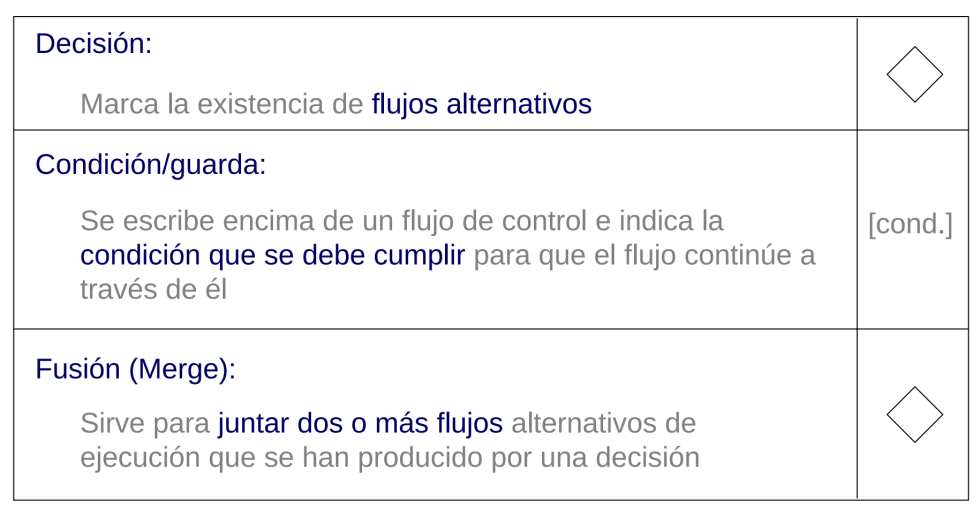
* Un proceso de negocio o un flujo de trabajo entre los usuarios y el sistema.
* Los pasos que se realizan en un caso de uso.
* Un protocolo de software, es decir, las secuencias de interacciones entre componentes permitidas.
* Un algoritmo de software.

**Elementos básicos**

Los elementos para representar los procesos que el sistema ejecuta se muestran en la figura x y en la figura y.

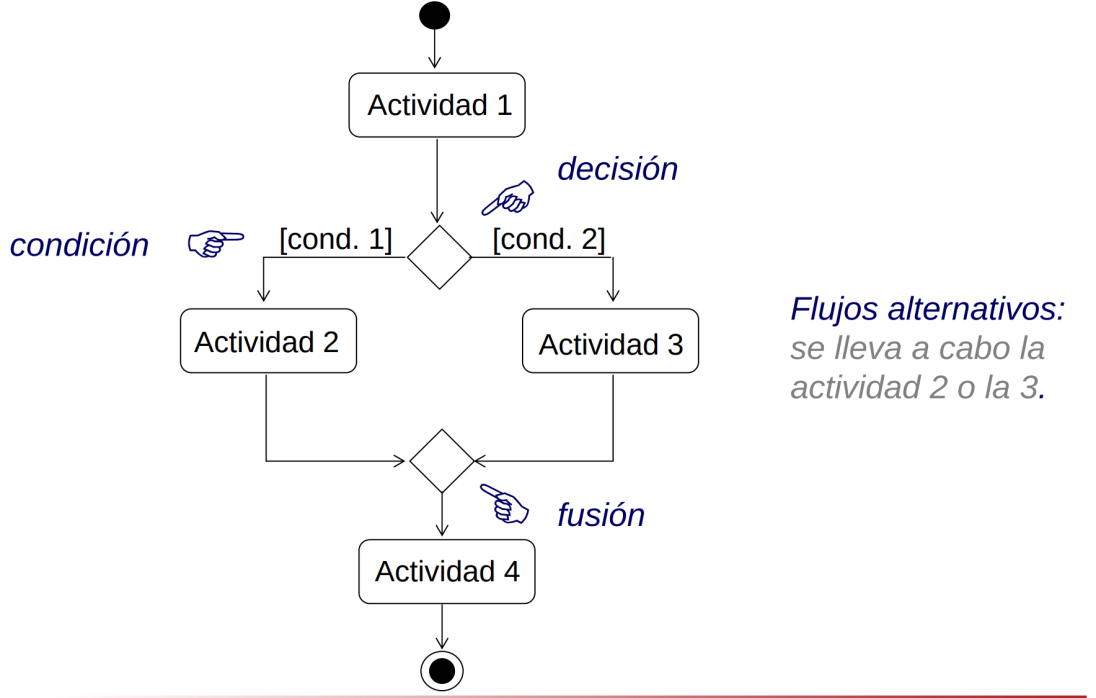


**Figura x.- Elementos de un diagrama de actividades.**

****

**Figura y.- Elementos decisivos de un diagrama de actividades.**

Un diagrama de actividades también permite representar flujos que ocurren de forma concurrente.En la **figura x** se muestra un ejemplo del diagrama de actividades con un flujo concurrente. La manera con la que se expresa concurrencia es con una linea horizontal y las actividades al mismo nivel.



**Figura x.- Diagrama de actividades concurrente**

**5.3.4.- Diagrama Relacional**

## 5.4.- Modelo Vista Controlador (MVC)

## 5.5.- Sistemas de Información

Se ha comentado anteriormente que un Sistema de Información hoy en día, es una parte fundamental en una Organización ya sea Publica o Privada, eso es un hecho ya que el avance y desarrollo tecnológico es evidente y la orientación hacia un mundo Globalizado e industrial esta haciendo que las empresas dependan mas de estos sistemas. Pero... ¿Que es un sistema de Información?.

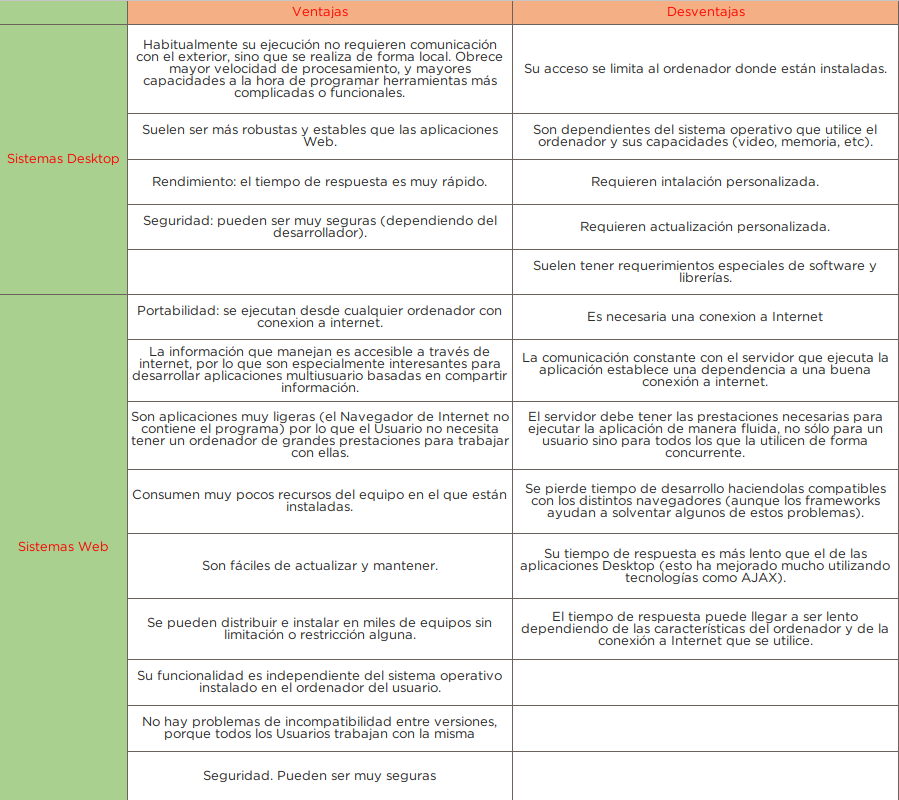
Un **Sistema Informático** se puede definir técnicamente como un conjunto de componentes relacionados que recolectan, procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de los procesos en una Organización (Laudon, K & Laudon, J, 2004).

Los Sistemas de información se basan en entornos **Desktop** o entornos **Web** de acuerdo a los requerimientos del negocio. Un Entorno **Desktop** ofrece un acceso local al sistema, permitiendo que el sistema opere y sea administrado dentro de la Organización, permitiendo que la información que se genera, se mantenga dentro de la misma Organización.

Sin embargo un Sistema **Web** es aquel sistema que se ejecuta sobre una un sistema dentro de un servidor en Internet o en una Intranet (red local). Dando así la libertad de accesar a ese sistema ya sea dentro o fuera de la empresa. Hoy en día los sistemas de Información orientados a Web son mas utilizados ya que ofrecen mas ventajas Sobre los sistemas Desktop. En la **Tabla 1** se observan las ventajas y desventajas de ambos sistemas.

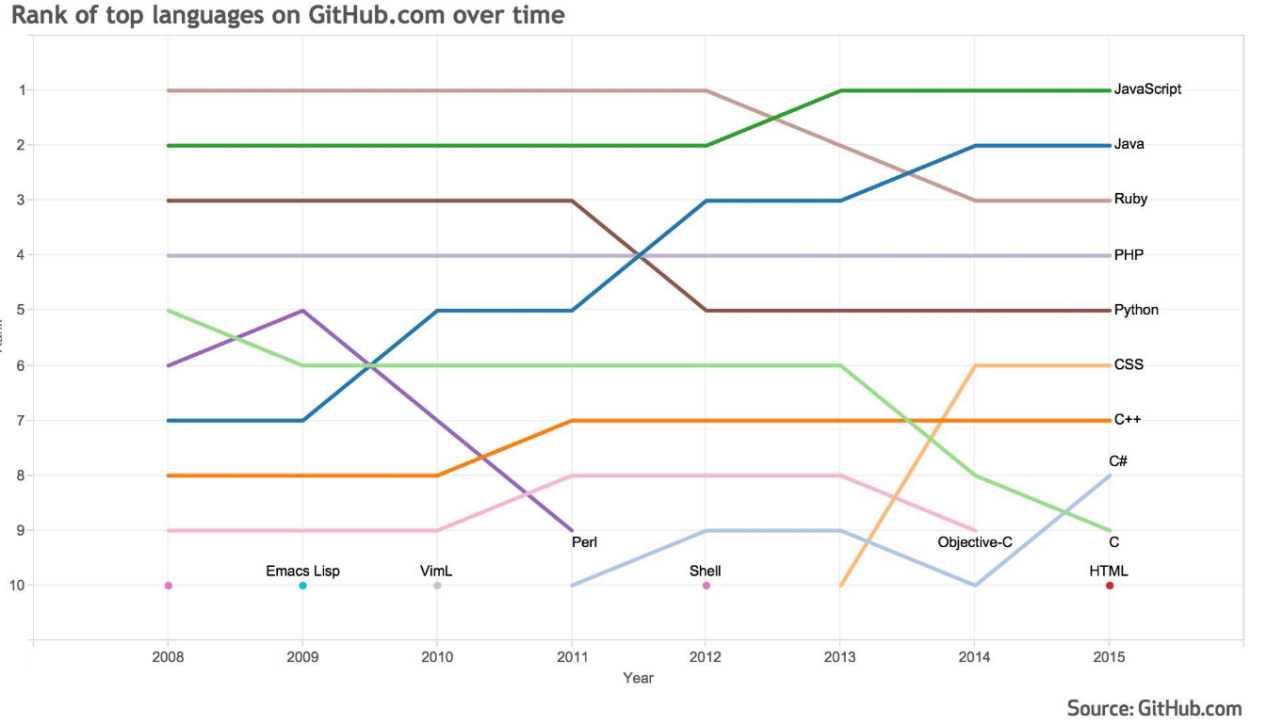
Todos los Sistemas Informáticos son programados en diferentes lenguajes de programación de acuerdo a las necesidades que pretenda cubrir.

Un **Lenguaje de Programación** consiste en un conjunto de Instrucciones y comandos que describen el proceso deseado. Cada lenguaje tiene sus instrucciones y enunciados verbales propios, que se combinan para formar los programas de computo. Los Lenguajes de Programación no son aplicaciones, sino herramientas que permiten construir y adecuar aplicaciones y sistemas de Información.



**Tabla 1.- Ventajas y desventajas de los Sistemas**

Según el ranking de los lenguajes mas utilizados en la plataforma Github desde su apertura, los 10 lenguajes mas utilizados para programar se muestran en el **Gráfico 1**. Como se puede Observar, **Java** es un lenguaje muy potente, adaptable, escalable y estable. Se puede programar sistemas tanto Desktop como Web, permitiendo así cubrir la mayoría de las necesidades que una Organización requiere en un sistema de Información.

**Gráfico 1. Ranking de Lenguajes mas utilizados.**

**Java** es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. El Objetivo clave de Java es poder escribir programas que se ejecuten en una gran variedad de sistemas Computacionales y dispositivos controlados por computadoras. En 1993 el servicio Web de Java se popularizó por la característica de agregar contenido dinámico y animaciones a las paginas Web. En la actualidad Java se utiliza para desarrollar aplicaciones y sistemas empresariales a gran escala, para mejorar la funcionalidad de los servidores Web (Paul Deitel y Harvey Deitel, 2012).

Un sistema Informático Web no funcionaria a menos que se tenga una Arquitectura de Sistema muy definida.

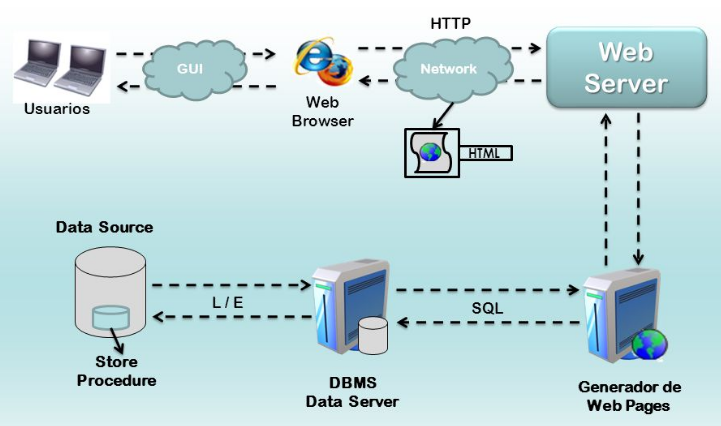
## 5.5.- Arquitectura de los Sistemas Web

Un sistema de Información basado en Web no seria mas eficiente que uno basado en Desktop si no tuviera una arquitectura de sistema perfectamente modelada, manteniendo la disponibilidad, integridad y confidencialidad tanto del sistema como de la información.

Un sistema de Información Web es proporcionado por un servidor Web y utilizada por usuarios que se conectan desde cualquier punto vía clientes Web (navegadores). La arquitectura de un Sitio Web tiene tres componentes principales:

1. Un servidor Web
2. Una conexión de red
3. Uno o más clientes

En la **Figura 1.** Se muestra el proceso petición (request) y respuesta (response) en un sistema de información Web. El cliente hace una petición remota al servidor, el servidor procesa la información, pasa por el gestor de base de datos, el gestor accede a los datos y los procesa esos datos para ser enviados de regreso. El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP. el servidor envía los datos por medio de este protocolo (response), el navegador del cliente procesa y renderiza la información por medio de HTML (HyperText Markup Language).

**Figura 1.- Arquitectura de un Sistema Web.**

La Arquitectura de un Sistema Web contiene diferentes componentes que interactúa entre si para procesar la información y brindar el un servicio, es importante manejar todos estos componentes para brindar un soporte adecuado a toda la arquitectura del sistema.

Como se muestra en la **figura x** los componentes básicos que conforman la arquitectura de los sistemas web son los siguientes:

1. Bases de Datos:
2. Administrador de Base de Datos
3. Servidor Web
4. Sistema de información
5. Red de acceso a internet o intranet
6. Navegador de Internet

# 6.- Desarrollo

A lo largo de la Residencia profesional se desarrolló un sistema de información al que se denomino “Sistema de Planeación Institucional de CECYTEM (SPIC)”, con el objeto de facilitar la disponibilidad, confidencialidad y la integridad de la información respecto a la información que el departamento de planeación va generando, se planteó desarrollar la primera fase del Sistema.

El sistema SPIC es un sistema web que se desarrolló en el Colegio de Estudios Científicos y Tecnológicos del Estado de México (CECYTEM), dependencia del sector de educación publica del Estado de México, del cual laboro como servidor publico.

La primera fase del sistema consiste en la gestión de las fichas técnicas de cada plantel que conforma el CECYTEM, este módulo del sistema pretende que el acceso a esta información sea rápida, organizada y en constante actualización.

El tiempo que se empleo para el desarrollo del sistema se contemplaron muchos aspectos técnicos y teóricos, acudiendo a capacitaciones y reuniones con compañeros para determinar la forma y manera con la que se trabaja, las tecnologías y el software que se ocupa en la infraestructura tecnológica del colegio, así como contemplar los procedimientos administrativos para la continuación del desarrollo del sistema.

En este informe técnico se detalla de manera especifica las actividades y las fases de desarrollo durante el desarrollo del sistema.

El sistema “SPIC” se basó sobre esta metodología basándose en los tiempos cortos de entregas, los requerimientos funcionales y el alcance que el sistema tiene, por otra parte la constante participación de los usuarios en la manera de diseñar y rediseñar la apariencia y la funcionalidad lógica del sistema.

## 6.1.- Análisis del Sistema

En el modelo de prototipado evolutivo cada fase del desarrollo del software, se recopila información para poder así afinar las características, funcionalidades y restricciones que el sistema alcanza como objetivo.

El Sistema de Planeación Institucional de CECYTEM (SPIC) surge de la necesidad de mantener, la información ordenada, disponible y de manera digital, para su consulta y actualización. El director del departamento de Planeación y Evaluación Institucional ordenó el digitalizar las fichas técnicas de los planteles. De acuerdo a este objetivo se llevo acabo el análisis de los requerimientos funcionales y no funcionales.

**Requerimientos funcionales y no funcionales**

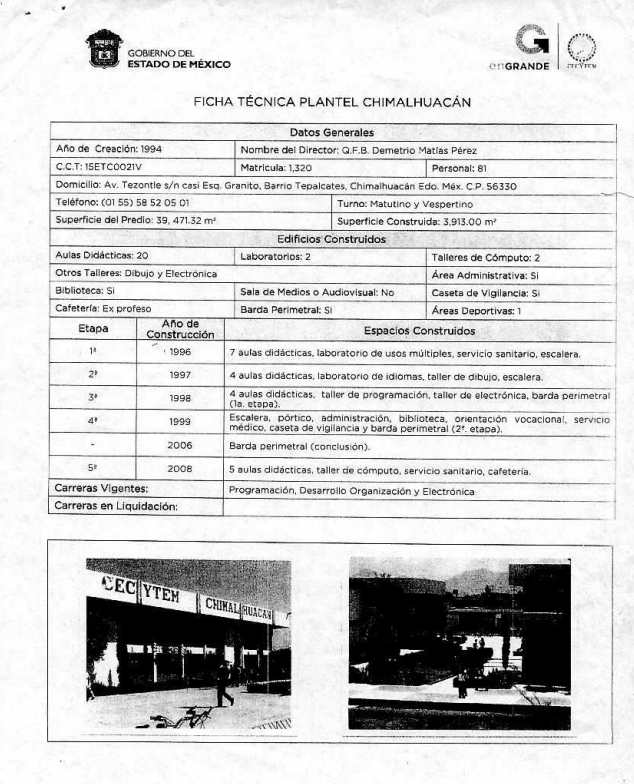
**Requerimientos funcionales**

1. El usuario podrá dar de alta una Ficha Técnica de Plantel
2. El usuario podrá modificar una Ficha Técnica de Plantel.
3. El Usuario podrá Imprimir una Ficha Técnica de un determinado plantel.
4. El Sistema de planeación podrá consultar información general y académica de los planteles desde el sistema "DEO".
5. La ficha Técnica gestionará las etapas de desarrollo del plantel y por tal podrá agregar, modificar y eliminar dichas etapas.
6. Se podrá subir y descargar evidencias (Fotos) de las etapas de los planteles.

**Requerimientos No Funcionales :**

1. El sistema tendrá acceso restringido a la información por medio de logueo de usuarios
2. Los usuarios tendrá diferentes roles dentro del sistema.
3. El sistema deberá estar disponible en cualquier momento, para su operación.
4. El sistema no tendrá restricción en cuanto a los usuario conectados.
5. Las impresiones de las Fichas Técnicas deberá adaptarse al tamaño de hoja carta.
6. El sistema deberá ser programado en JSP de Java.
7. El Sistema deberá trabajar con una base de datos SQL Server.
8. El sistema se apegará a los estándares de desarrollo del colegio.

Como se Observa en la **figura x** el formato de la ficha técnica que se manejaba, un requerimiento del departamento fue extender los campos de información añadiendo los campos de geolocalización (altitud, latitud), personal administrativo, docentes.

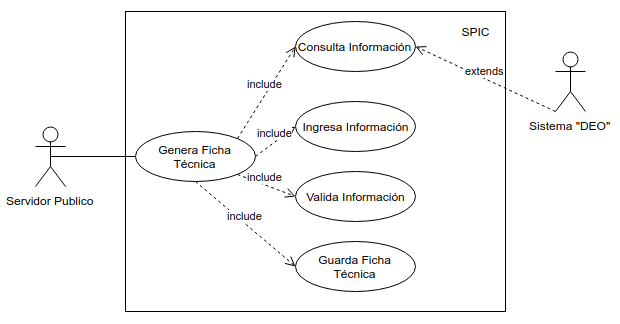
Al haber entendido los requerimientos fundamentales se realizo el diseño de la aplicación web.

## 6.2.- Diseño de la aplicación web SPIC

**Diagramas de Casos de uso**

Se realizaron una serie de casos de uso para definir los procesos y funciones que el sistema debe de replicar.

En la **figura x** se muestra un diagrama de caso de uso que se ocupó para el diseño del sistema. En el se muestra el caso de uso para generar una nueva ficha técnica de plantel, se generan cuatro procesos que se incluyen en el caso principal, el subcaso “”consulta información” extiende del sistema “DEO”.



**Figura x. Caso de Uso de “Generar ficha técnica”.**

En el **Anexo A** se encuentran los diagramas de todos los casos de uso del sistema.

**Diagrama de Actividades**

**Diagrama de Clases**

**Diagrama Relacional**

## 6.3.- Desarrollo de la aplicación web SPIC

En el desarrollo de la aplicación SPIC, se trabajó con diferentes herramientas para dar solución a los requerimientos establecidos por el departamento de Planeación y Evaluación Institucional, el sistema se adapto de manera que fuera compatible con diferentes aspectos de los sistemas “DEO” y “SIBIEN” y los estándares de desarrollo del CECYTEM.

**6.3.1 Tecnologías Utilizadas**

Se utilizaron diferentes tecnologías que en conjunto trabajan para dar funcionalidad al sistema SPIC, A continuación se describen de manera detallada.

1. SQL SERVER 2016: el sistema gestor de base de datos de Microsoft, son las estructuras fuente de toda la información de los sistemas que se ocupan en el CECYTEM, el sistema “SPIC” trabaja con este gestor.
2. Sistema operativo Windows Server 2008 R2, la versión de este sistema, especializado en servidores, con el se opera a niveles base de datos y aplicaciones en el CECYTEM.
3. Servidor de Aplicaciones Tomcat Apache 8.0.27, el servidor de aplicaciones que se ocupó a nivel desarrollo a lo largo del proyecto.
4. Servidor de aplicaciones Wildfly 10.0.1, al concluir el desarrollo del sistema “SPIC” se liberó la primera versión, la cual se despliega en este servidor, cabe mencionar que se utiliza para el despliegue de aplicaciones a nivel producción en el CECYTEM.
5. JRE 8.12, maquina virtual para ejecutar aplicaciones java, se ocupó a lo largo del desarrollo.
6. Java, el lenguaje de programación que se ocupó para el desarrollo del sistema “SPIC”.
7. JSP, tecnología java que en conjunto con HTML le dio funcionalidad y optimización a la manera de programar el sistema “SPIC”.
8. HTML5, con esta tecnología se programaron las paginas web del sistema.
9. CSS3, tecnología que se utilizó para darle presentación y estética a las paginas web del sistema.
10. JavaScript, ayudo al despliegue de notificaciónes y mensajes en el sistema.
11. Navegador Chrome 52.0.27, con las herramientas de desarrollo integradas en el navegador.
12. Librerias: JQuery 2.0.3, Commons de Apache 1.2.2, JDBC 4.2, JDK 4.8.
13. XML para la configuración de archivos del servidor de aplicaciones.

**6.3.1 Modelo de Desarrollo**

La lógica del sistema “SPIC”, al igual que los sistemas “DEO” y “SIBIEN”, trabajan con el Modelo Vista Controlador (MVC), como en la **sección x** se explicó la arquitectura de este modelo, el sistema “SPIC” recibe y responde por medio de una clase llamada “controller.java”, en esta clase es el núcleo de la funcionalidad y el procesamiento de información para el usuario y para el mismo sistema. En la **figura x** se muestra la lógica del sistema al implementar este modelo.

Figura.

# 7.- Resultados

# 8.- Conclusiones

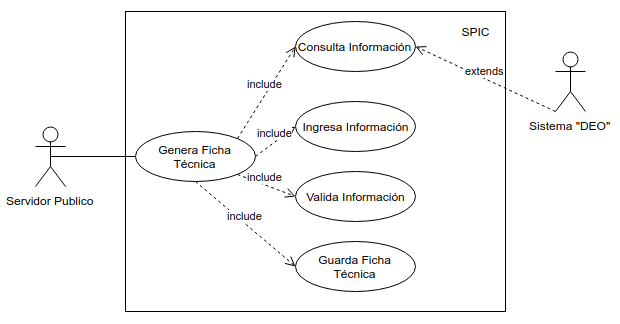
# 9.- Referencias

1. Shari Lawrence Pfleeger. (2002). Ingeniería de Software. Argentina: Prentice Hall .
2. Ian Sommerville. (2005). Ingeniería del Software. España: Pearson.
3. Roger S. Pressman. (2010). Ingeniería del Software. Un enfoque Practico. México: Mc Graw Hill.
4. Laudon, K & Laudon, J. (2004). Sistemas de Información. Octava Edición. México: Editorial Pearson - Prentice Hall.

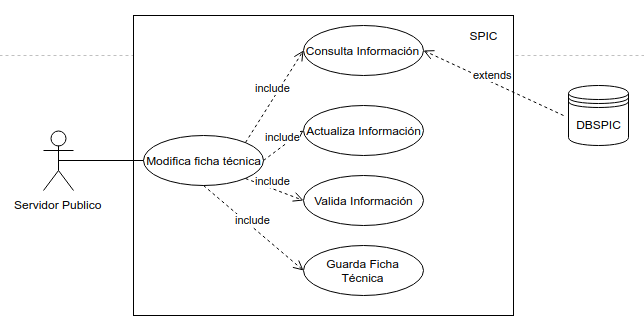
# - Anexos

## 10.1 Anexo A

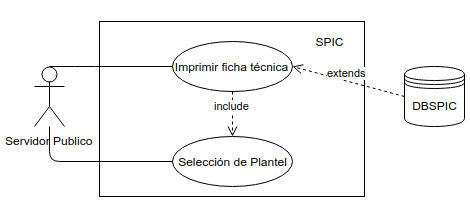
1. Diagrama de casos de uso del Sistema SPIC
2. Generar ficha técnica.

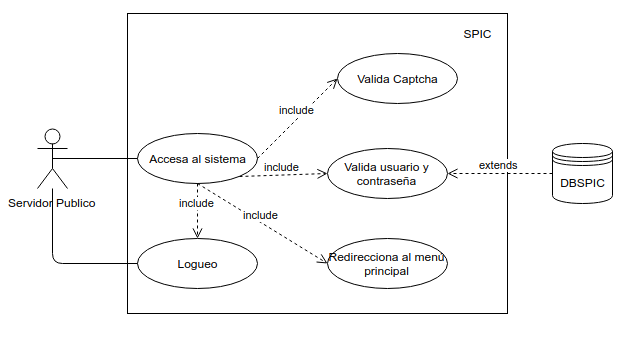


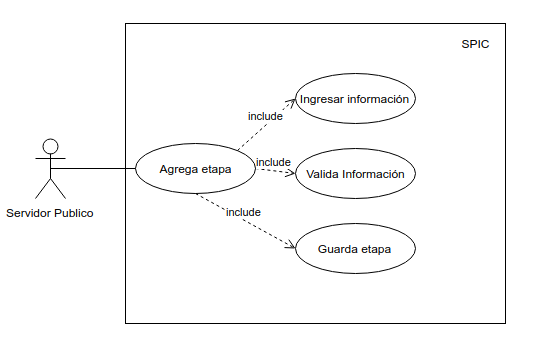
1. Modificar ficha técnica.



1. Imprimir ficha técnica.



1. Acceso al sistema SPIC.
2. Agregar etapa de desarrollo.



1. Modificar ficha técnica.

