



**ESTÁNDAR TÉCNICO DE COMPETENCIAS PARA
EL DESARROLLO DE SOFTWARE**

INGENIERO DE SOFTWARE

'TALENTO EN TI'

ABRIL DE 2014

ÍNDICE DEL CONTENIDO

- 1 OBJETIVO**
- 2 CAMPO DE APLICACIÓN**
- 3 DEFINICIONES**
- 4 REQUISITOS DEL PERFIL**
- 5 BIBLIOGRAFÍA**
- 6 LISTA DE MENCIONES**

EVALUACIÓN DE INGENIERO DE SOFTWARE

1 OBJETIVO

El presente estándar técnico tiene por objeto establecer los requisitos que se deben cubrir para obtener la certificación como Ingeniero de Software.

2 CAMPO DE APLICACIÓN

Este estándar técnico debe aplicarse para evaluar, calificar y certificar aspirantes a ser “Ingeniero de Software” en el marco establecido por el proyecto TALENTO EN TI¹ en los niveles de Junior y Senior.

También debe aplicarse para evaluar el mantenimiento de la competencia de las personas que hayan sido certificadas como “Ingeniero de software”, en cualquiera de sus niveles.

Este estándar también puede aplicarse como guía para el desarrollo y definición de programas de formación académica y de capacitación profesional.

2.1 Justificación

Derivado de la necesidad de definir un perfil específico para la Ingeniería de software acorde a los requerimientos de la Industria Mexicana de Tecnologías e Información y Comunicación.

Las Instituciones de educación podrán utilizar el presente estándar como referente para la creación de programas y/o adaptación de su retícula; a la Industria como referente para la selección y contratación de personal competente para cubrir sus necesidades de Ingeniería de Software; y para el aspirante como referente de los requisitos necesarios para obtener la certificación como “Ingeniero de Software”

3 DEFINICIONES Y ABREVIATURAS

Las definiciones y abreviaturas aplicables a este estándar son las siguientes:

3.1 Ingeniero de software. Es el profesional encargado de los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema hasta el mantenimiento, maximizando la calidad y optimizando recursos.

3.2 Rastreabilidad de requisitos de software. Es la validación de los requerimientos iniciales del sistema contra el producto final.

3.3 Framework. Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.²

¹ Anexo A, del convenio específico de colaboración Celebrado entre la Asociación Academia, Industria, Gobierno en Tecnologías de la Información, A.C. (IMPULSA-TI) y Normalización y Certificación Electrónica (NYCE, A.C.), suscrito el 15 de Septiembre de 2010.

² <http://es.wikipedia.org/wiki/Framework>

INGENIERO DE SOFTWARE

ETCP-IS

3.4 Arquitectura de software. Es el proceso por el cual se propone una solución para atender los requisitos técnicos y operacionales de un sistema de información. Define qué componentes conforman el sistema, cómo se relacionan entre ellos y cómo mediante su interacción cumplen con la funcionalidad deseada, considerando los criterios de calidad establecidos y lo riesgo inherentes al sistema y su entorno.

3.5 Arquitecto de Software. Es un profesional que establece la estructura de diseño y lineamientos de calidad tanto para ingenieros, desarrolladores de software y la infraestructura que soporta los sistemas de información, para que trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades del cliente, optimizando el uso de recursos y frameworks.

3.6 Sistema de información. Es un conjunto de elementos informáticos (equipo de telecomunicaciones o computacional, software, infraestructura y personas) que trabajan en conjunto, teniendo como objetivo obtener, almacenar, manipular, administrar, mover, controlar, desplegar, intercambiar, transmitir o recibir datos, e incluye tanto los programas de computación ("software" y "firmware") como el equipo de cómputo.

3.7 Portafolio de evidencias. Se integra por las evidencias de desempeño y las evidencias por producto que se determinen como entregables antes de la examinación en el centro de evaluación. Las evidencias que se exijan como parte del Portafolio de Evidencias será, antecedidas por el acrónimo PE

3.8 Experiencia. Actividades para las cuales el profesional debe demostrar que cuenta con la capacidad de llevarlas a cabo,

3.9 Actividades prácticas. Entregables que deben ser elaborados por el aspirante demostrar competencia en este elemento

3.10 Conocimiento. Reactivos correspondientes a la evaluación cuantitativa de este elemento

3.11 Mantenimiento de la competencia. Mantenimiento de la competencia: Actividades realizadas para supervisar que el sustentante de un certificado sigue cumpliendo con los requisitos marcados por el Estándar Técnico y que su desempeño ha sido acorde con el código de ética vigente

3.12 Retícula Es una forma de presentar la información, son guías para la alineación y la distribución de los elementos en un formato. Organiza todos los elementos y los jerarquiza, creando un orden intangible que puede ser comprendido y repetido por las personas. Para proyectos complejos se emplea la retícula conocida como modular la cual consiste en columnas con un gran número de líneas de flujo horizontales que subdividen las columnas en fila, creando una matriz de celdas denominadas módulos.

<i>Fecha de Aprobación</i>	<i>Fecha de Publicación</i>	<i>Tiempo en que deberá revisarse</i>
<i>30 de abril de 2014</i>	<i>15 de mayo de 2014</i>	2 años después de la fecha de publicación.

4 REQUISITOS DEL PERFIL

Prerrequisitos para Ingeniero de Software

NIVEL JUNIOR	NIVEL SENIOR
- Pasante o Estudiante del último año de la licenciatura de TI	- Título de licenciatura o - 5 años de experiencia en el mercado de desarrollo de software

Los requisitos básicos que debe demostrar un profesional como candidato para certificarse como Ingeniero de Software son los siguientes:

NIVEL JUNIOR	NIVEL SENIOR
<ol style="list-style-type: none"> 1. Capacidad crítica. 2. Control de cambios. 3. Solución de problemas 4. Trabajo en equipo. 	<ol style="list-style-type: none"> 1. Análisis y Síntesis de información. 2. Capacidad crítica. 3. Comunicación Oral y Escrita. 4. Control de cambios. 5. Solución de problemas 6. Autodidacta 7. Redacción de informes 8. Trabajo en equipo.

Cada uno de los requisitos cuenta con “Experiencia” que deberá estar soportada por la presentación de registros documentados de la experiencia profesional del candidato. Las “Actividades prácticas” deberán ser soportadas por los resultados de la resolución de un caso práctico asignado o desarrollado por el aspirante. Los “Conocimientos” serán cubiertos mediante la presentación de un examen de conocimientos, mismos que se encuentran descritos dentro de cada uno de los elementos listados. Las evidencias solicitadas deben ser entregadas en su totalidad para aspirar a una evaluación satisfactoria.

Asimismo, las competencias específicas que distinguen a un Ingeniero de Software son las siguientes:

NIVEL JUNIOR	NIVEL SENIOR
<ol style="list-style-type: none"> 1. Algoritmos y Estructuras de Datos 2. Manejo de metodologías y estándares para desarrollo de software 3. Conocimiento y codificación en lenguajes de programación 4. Análisis de requerimientos de Software 5. Especificaciones de diseño de software 6. Diseño de interfaces de software 7. Pruebas de Software 8. Seguridad del software 9. Integración de componentes de Software 10. Bases De Datos 11. Mantenimiento de Software 12. Nuevas Tecnologías 	<ol style="list-style-type: none"> 1. Algoritmos y Estructuras de Datos 2. Manejo de metodologías y estándares para desarrollo de software 3. Conocimiento y codificación en lenguajes de programación 4. Análisis de requerimientos de Software 5. Especificaciones de diseño de software 6. Diseño de interfaces de software 7. Pruebas de Software 8. Seguridad del software 9. Integración de componentes de Software 10. Bases De Datos 11. Mantenimiento de Software 12. Transferencia de conocimiento y documentación de software 13. Nuevas Tecnologías

Cada una de las competencias cuenta con “Experiencia” que deberá estar soportada por la presentación de registros documentados de la experiencia profesional del candidato. Las “Actividades prácticas” deberán ser soportadas por los resultados de la resolución del caso práctico asignado o desarrollado por el aspirante. “Los Conocimientos” serán cubiertos mediante la presentación de un examen de conocimientos, mismos que se encuentran descritos dentro de cada uno de los elementos listados. Las evidencias solicitadas deben ser entregadas en su totalidad para aspirar a una evaluación satisfactoria.

4.1 Criterios de evaluación Genéricos

ELEMENTO 1: Análisis y Síntesis de información.

NIVEL JUNIOR	NIVEL SENIOR
	<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Redactar un documento con las necesidades generales del proceso que se le ha asignado en el caso práctico y además un párrafo que indique el objetivo del esfuerzo para resolver el problema. Indique el objetivo del esfuerzo para resolver el problema especificando el tiempo necesario. • Redactar un documento en el cual se propongan dos posibles soluciones al caso práctico.

ELEMENTO 2: Capacidad crítica.

NIVEL JUNIOR	NIVEL SENIOR
<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Realizar un juicio sobre la elección entre las dos alternativas de solución propuestas para el caso práctico, del elemento 1: análisis y síntesis de información. El candidato deberá justificar su elección. <p>Conocimiento</p> <ul style="list-style-type: none"> • Conocimiento de técnicas de apoyo para toma de decisiones. <ul style="list-style-type: none"> -Lluvia de ideas -Árboles de decisiones -Diagrama de causa-efecto -Diagrama de flujo -Ponderaciones • Conocimiento del proceso de toma de decisiones <ul style="list-style-type: none"> -Identificación del problema -Análisis del problema (Causas, impacto, prioridad de resolución) -Determinación de soluciones posibles -Determinación de criterios de decisión 	<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Realizar un juicio sobre la elección entre las dos alternativas de solución propuestas para el caso práctico, del elemento 1: análisis y síntesis de información. El candidato deberá justificar su elección. <p>Conocimiento</p> <ul style="list-style-type: none"> • Conocimiento de técnicas de apoyo para toma de decisiones. <ul style="list-style-type: none"> ○ Lluvia de ideas ○ Árboles de decisiones ○ Diagrama de causa-efecto ○ Diagrama de flujo ○ Ponderaciones • Conocimiento del proceso de toma de decisiones <ul style="list-style-type: none"> ○ Identificación del problema ○ Análisis del problema (Causas, impacto, prioridad de resolución) ○ Determinación de soluciones posibles ○ Determinación de criterios de decisión

INGENIERO DE SOFTWARE ETCP-IS

<ul style="list-style-type: none"> -Evaluación de las soluciones -Elección de la mejor solución o soluciones 	<ul style="list-style-type: none"> o Evaluación de las soluciones o Elección de la mejor solución o soluciones
--	--

ELEMENTO 3: Comunicación Oral y Escrita.

NIVEL JUNIOR	NIVEL SENIOR
	<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Evidencia ELEMENTO 1. Se tomará de base la evidencia generada en ese ELEMENTO para revisar la correcta redacción y la ortografía. <ul style="list-style-type: none"> • Proceso de la comunicación • Lenguaje •

ELEMENTO 4: Control de cambios.

NIVEL JUNIOR	NIVEL SENIOR
<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Presentar un formato lleno para requisición de cambios donde evalúa el impacto, para el cambio presentado en el Caso Práctico. <p>Evidencia de conocimiento</p> <ul style="list-style-type: none"> • Análisis de viabilidad • Análisis de impacto 	

ELEMENTO 5: Solución de problemas

NIVEL JUNIOR	NIVEL SENIOR
<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Documentar el Análisis de causa raíz, mencionando la metodología utilizada, definir y decidir sobre dos alternativas de solución para un problema específico presentado en el caso práctico, durante la ejecución del proyecto. <p>Evidencia de conocimiento</p> <p>Conocimiento sobre alguna metodología de análisis de problemas</p>	

ELEMENTO 6: Autodidacta

NIVEL JUNIOR	NIVEL SENIOR
	<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Generar una lista de términos y definiciones que aprendió o profundizó durante la documentación del caso práctico, incluyendo la

INGENIERO DE SOFTWARE

ETCP-IS

	bibliografía. Se deberán omitir aquellos términos que sean básicos en la terminología de la Ingeniería de Software actuales y que ya han sido aprendidos por el aspirante previamente.
--	--

ELEMENTO 7: Redacción de informes

NIVEL JUNIOR	NIVEL SENIOR
	Actividades prácticas <ul style="list-style-type: none">Redactar un informe de avance sobre la solución del problema que se le asignó en el caso Práctico, con los detalles del componente desarrollado y las actividades pendientes por realizar. Tomando en cuenta que se lleva un 60% de avance. Es decir, indicar lo que se ha realizado, con su estado actual y qué está pendiente.

ELEMENTO 8: Trabajo en equipo

NIVEL JUNIOR	NIVEL SENIOR
Conocimiento <ul style="list-style-type: none">Técnicas de trabajo en equipo	

4.2 Criterios de evaluación Específicos

ELEMENTO 1: Algoritmos y Estructuras de Datos

NIVEL JUNIOR	NIVEL SENIOR
Actividades prácticas <ul style="list-style-type: none">Presentación de un algoritmo en pseudocódigo de ordenamiento de acuerdo al ejercicio asignado, que se presenta al final del estándar con el título "propuesta de ejercicio para ordenamiento de nombres". Explicar su funcionamiento y explicar su elección.	
Conocimiento <ul style="list-style-type: none">Algoritmos computacionales<ul style="list-style-type: none">OrdenamientoBúsquedasRecursividadDivisión (para manejo óptimo de recursos computacionales)ArreglosListasÁrbolesTablas hash	

ELEMENTO 2: Manejo de metodologías y estándares para desarrollo de software

NIVEL JUNIOR	NIVEL SENIOR
<p>Conocimiento</p> <ul style="list-style-type: none"> • UML • Diseño en cascada • Ciclo de vida de software • MoProSoft (Administración de Proyectos de Software, Desarrollo y Mantenimiento de software) • CMMI (REQM - Gestión de requerimientos, RD – Desarrollo de requerimientos, Entendimiento de los niveles de madurez) 	

ELEMENTO 3: Conocimiento y codificación en lenguajes de programación

NIVEL JUNIOR	NIVEL SENIOR
<p>Experiencia</p> <ul style="list-style-type: none"> • Evidencias de al menos 40 horas de capacitación en algún lenguaje de programación. <p>Actividades prácticas</p> <ul style="list-style-type: none"> • Generar un módulo o los módulos necesarios para demostrar la funcionalidad completa de alguno de ellos. Incluyendo el código fuente, de acuerdo a un caso práctico. <p>Conocimiento</p> <ul style="list-style-type: none"> • Lenguajes de Programación (C y C#) • Estructuras básicas de programación <ul style="list-style-type: none"> ○ Iteración ○ Anidación ○ Bifurcación <ul style="list-style-type: none"> ▪ Simple ▪ Múltiple ○ Secuencial • Tipos de datos simples y estructurados • Funciones y métodos • Compiladores intérpretes y traductores • Frameworks 	<p>Experiencia</p> <ul style="list-style-type: none"> • Evidencias de al menos 40 horas de capacitación en algún lenguaje de programación. <p>Actividades prácticas</p> <ul style="list-style-type: none"> • Generar un módulo o los módulos necesarios para demostrar la funcionalidad completa de alguno de ellos. incluyendo código fuente, de acuerdo al caso práctico. <p>Conocimiento</p> <ul style="list-style-type: none"> • Lenguajes de Programación (Java, NET, Cobol, RPG, PLI, etc.). • Estructuras básicas de programación. <ul style="list-style-type: none"> ○ Iteración ○ Anidación ○ Bifurcación <ul style="list-style-type: none"> ▪ Simple ▪ Múltiple ○ Secuencial • Tipos de datos simples y estructurados • Funciones y métodos • Compiladores intérpretes y traductores • Frameworks.

ELEMENTO 4: Análisis de requerimientos de Software

NIVEL JUNIOR	NIVEL SENIOR
<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Generar Especificación de Requerimientos de Software • Generar Caso de Uso de acuerdo al caso práctico o especificación de análisis. 	

<p>Conocimiento</p> <ul style="list-style-type: none"> • Análisis de requerimientos de software para soluciones técnicas • Casos de uso
--

ELEMENTO 5: Especificaciones de diseño de software

NIVEL JUNIOR	NIVEL SENIOR
<p>Actividades prácticas</p> <p>Generar artefactos de diseño de acuerdo al caso práctico</p> <ul style="list-style-type: none"> • Generar Diagrama de Clases. • Generar Diagrama de Secuencia de acuerdo al caso práctico. • Generar Diagrama de Colaboración de acuerdo al caso práctico. • Diseño de interfaz gráfica. • Diagrama de componentes de la solución y su relación. <p>Conocimiento</p> <ul style="list-style-type: none"> • Diagramas de Clases • Diagramas de Secuencia. • Diagramas de Estados. • Diagramas de Colaboración. • Especificaciones de diseño para sistemas tradicionales o modernos • Interfaz gráfica 	

ELEMENTO 6: Diseño de interfaces de software

NIVEL JUNIOR	NIVEL SENIOR
<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Generar especificación de interfaces de software de acuerdo al caso práctico. <p>Conocimiento</p> <ul style="list-style-type: none"> • Generar especificación de interfaces de software de acuerdo al caso práctico, estas interfaces deben ser con componentes/aplicaciones externas a la aplicación principal que de la solución. 	

ELEMENTO 7: Pruebas de Software

NIVEL JUNIOR	NIVEL SENIOR
<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Generar matriz de pruebas (con algunos casos) de acuerdo al caso práctico, presentando las diferentes alternativas comunes de verificación y validación de requisitos, así como el resultado esperado de la aplicación de los casos.. 	

<p>Conocimiento</p> <ul style="list-style-type: none"> • Técnicas de pruebas de requisitos de software. <ul style="list-style-type: none"> ○ Funcional ○ Análisis de valores límite ○ De volumen ○ Estrés ○ Validación de datos ○ Inyección de defectos de software y hardware ○ De desempeño (uso eficiente de recursos)

ELEMENTO 8: Seguridad del software

NIVEL JUNIOR	NIVEL SENIOR
<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Generar diseño básico de seguridad para acceso a la aplicación de acuerdo al caso práctico. <p>Conocimiento</p> <ul style="list-style-type: none"> • Esquemas de seguridad de acceso a aplicaciones. <ul style="list-style-type: none"> ○ Métodos de acceso (usuario, password, grupos, perfiles) ○ Métodos de criptografía (encryption) <ul style="list-style-type: none"> ▪ Algoritmos de Criptografía simétricos ▪ Algoritmos de Criptografía asimétricos ○ Algoritmos de Hash 	

ELEMENTO 9: Integración de componentes de Software

NIVEL JUNIOR	NIVEL SENIOR
<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Generar plan secuencia de integración de componentes de acuerdo al caso práctico. Esto es, una lista de componentes que serán integrados, ambiente (hardware y software) en donde se integrarán para pruebas y producción. <p>Conocimiento</p> <ul style="list-style-type: none"> • integración de componentes 	

ELEMENTO 10: Bases De Datos

NIVEL JUNIOR	NIVEL SENIOR
<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Generar diagrama entidad-relación de acuerdo al caso práctico. <ul style="list-style-type: none"> ○ Entidades ○ Relaciones • Normalizar diagrama entidad-relación hasta el 3er. Nivel 	

INGENIERO DE SOFTWARE ETCP-IS

<ul style="list-style-type: none"> • Generar Índices (primarios y secundarios o foráneos) • Generar scripts básicos para generación de esquema físico de Base de Datos <p>Conocimiento</p> <ul style="list-style-type: none"> • Diseño de Base de Datos • Tipos de Bases de Datos
--

ELEMENTO 11: Mantenimiento de Software

NIVEL JUNIOR	NIVEL SENIOR
<p>Conocimiento</p> <ul style="list-style-type: none"> • Lenguaje de Programación (C y C#) • Frameworks 	<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Los siguientes ELEMENTOS se tomarán como base para determinar si el candidato puede ser capaz de realizar actividades de Mantenimiento de Software <ul style="list-style-type: none"> ○ Evidencia del ELEMENTO 3, Codificación en lenguajes de programación. ○ Evidencia del ELEMENTO 2 Capacidad crítica <p>Conocimiento</p> <ul style="list-style-type: none"> • Lenguaje de Programación (Java, .NET, Cobol, RPG, PLI, etc.). • Frameworks.

ELEMENTO 12: Transferencia de conocimiento y documentación de software

NIVEL JUNIOR	NIVEL SENIOR
	<p>Actividades prácticas</p> <ul style="list-style-type: none"> • Presentación en diapositivas de la funcionalidad del componente desarrollado en el caso práctico para que sea entendido por los usuarios del sistema. • Documentación del código fuente desarrollado para el componente del caso práctico ya que usualmente quien proporciona el Mantenimiento necesita identificar a través del código los puntos a modificar. <p>Conocimiento</p> <ul style="list-style-type: none"> • Estándares de documentación de código fuente

ELEMENTO 13: Nuevas Tecnologías

NIVEL JUNIOR	NIVEL SENIOR
Conocimiento <ul style="list-style-type: none">• Desarrollo de aplicaciones para dispositivos móviles• Cómputo en la nube• Virtualización<ul style="list-style-type: none">○ Como herramienta de apoyo en la fase de pruebas	

Caso práctico Incluye:

- Descripción de la problemática general
- Descripción de la organización ficticia
- Requerimientos del cliente
- Definición de un problema durante la ejecución del proyecto que implique decidir entre dos alternativas para resolver éste (definición de un problema que implique un análisis de causa raíz y decisión de una alternativa para resolverla)
- Definición de un cambio en los requerimientos iniciales

Propuesta de ejercicio para ordenamiento de nombres (Evidencia por producto del elemento 1 “Algoritmos y estructuras de Datos” de la sección 4.2.

Se cuenta con un archivo de texto plano el cual contiene 10,000 nombres de clientes, la cual debe ser ordenada alfabéticamente.

5 BIBLIOGRAFÍA

Fundamentos de Programación, algoritmos, estructura de datos y objetos, Luis Joyanes Aguilar, Editorial Mc Graw Hill
Ingeniería de Software, teoría y práctica, Shari Lawrence Pfleeger, Prentice Hall.
Ingeniería de Software, Ian Sommerville, 6ª edición, Addison Wesley.
Ingeniería de Software, Una perspectiva orientada a objetos, Eric J. Braude, Alfaomega.
Fundamentos de programación, Luis Joyanes Aguilar
Manual de certificación Sun Services
Object- Oriented Analysis and Design Using UML, Student Guide
NMX-I-27001-NYCE. Sistemas de gestión de la seguridad de la información
NMX-I-20000-NYCE. Sistemas de Gestión de servicios de tecnología de la información.
(Ingeniería de software, Ian sommerville, 6ª edición)
Fundamentos de base de datos. Adoración de Miguel, Mario Piattini. AlfaOmega

6 LISTA DE MENCIONES

El presente estándar técnico fue elaborado en la participación de:

- SOCIEDAD ACADEMIA INDUSTRIA Y GOBIERNO EN TECNOLOGÍAS DE LA INFORMACIÓN (IMPULSA-TI)
- UNIVERSIDAD AUTÓNOMA METROPOLITANA (AZCAPOTZALCO)

INGENIERO DE SOFTWARE ETCP-IS

- TECNOLÓGICO DE ESTUDIOS SUPERIORES DE ECATEPEC
- INFORMÁTICA INTEGRAL EMPRESARIAL, S.A. DE C.V.
- OKYSOFT, S.A. DE C.V.
- TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CUAUTITLÁN IZCALLI
- FACULTAD DE ESTUDIOS SUPERIORES DE CUAUTITLÁN IZCALLI
- FACULTAD DE ESTUDIOS SUPERIORES DE ACATLÁN
- INNEVO
- CENTRO DE INGENIERÍA DE SOFTWARE I + D (UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ)
- NORMALIZACIÓN Y CERTIFICACIÓN ELECTRÓNICA, A.C.
- UNIVERSIDAD POPULAR AUTÓNOMA DEL ESTADO DE PUEBLA. (UPAEP)
- ASOCIACIÓN NACIONAL DE INSTITUCIONES DE EDUCACIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN