

**IDENTIFICACIÓN DE MEJORES PRÁCTICAS DE CALIDAD PARA EL
DESARROLLO DE SOFTWARE PARA SER IMPLEMENTADAS EN LOS
PROCESOS DE FORMACIÓN DE LA MEDIA TÉCNICA EN PROGRAMACIÓN -
ALIANZA FUTURO DIGITAL MEDELLIN**

CLAUDIA MARCELA RAQUEJO ÁLVAREZ

**Proyecto de grado para optar el título de Maestría en Ingeniería con
especialidad en Tecnologías de Información para educación**

Asesor:

MARÍA DEL ROSARIO ATUESTA VENEGAS

**UNIVERSIDAD EAFIT
ESCUELA DE INGENIERÍAS
MAESTRÍA EN INGENIERÍA
ÉNFASIS EN TECNOLOGÍAS DE INFORMACIÓN PARA LA EDUCACIÓN
MEDELLÍN
2015**

DEDICATORIA

A mis estudiantes que son la razón de SER maestra, que con su alegría y vitalidad me brindan la energía para seguir adelante, amar y realizar mi labor.

A mi familia que me permitió su tiempo para dedicarme a este proyecto.

AGRADECIMIENTOS

A Dios por permitirme recorrer el camino de la Educación de jóvenes.

A la Secretaria de Educación de Medellín y a la Alianza Futuro Digital Medellín que facilitó este proceso de formación.

Para mis maestros y compañeros de maestría, que con sus aportes y discusiones permitieron desarrollar esta propuesta para la mejora de los procesos educativos.

A María del Rosario, mi asesora que con su profesionalismo, experiencia y visión, me guió eficientemente en este proceso de investigación.

TABLA DE CONTENIDO

<u>1</u>	<u>INTRODUCCIÓN.....</u>	<u>1</u>
<u>2</u>	<u>PLANTEAMIENTO DEL PROYECTO.....</u>	<u>3</u>
<u>3</u>	<u>DESCRIPCIÓN DEL PROBLEMA.....</u>	<u>4</u>
<u>4</u>	<u>ESTADO DEL ARTE (ANTECEDENTES)</u>	<u>17</u>
<u>5</u>	<u>OBJETIVOS</u>	<u>19</u>
5.1	OBJETIVO GENERAL.....	19
5.2	OBJETIVOS ESPECÍFICOS	19
<u>6</u>	<u>MARCO DE REFERENCIA</u>	<u>20</u>
6.1	PROCESO DE DESARROLLO DE SOFTWARE.....	20
6.2	CALIDAD EN EL PROCESO DEL SOFTWARE.....	21
6.3	MODELOS DE REFERENCIA PARA LA CALIDAD EN EL PROCESO.....	26
6.4	CALIDAD DEL SOFTWARE RELACIONADA CON EL PRODUCTO	28
6.5	CALIDAD RELACIONADA CON LAS PERSONAS	31
6.5.1	PSP/TSP	32
6.5.2	SCRUM.....	34
6.5.3	METODOLOGÍA XP PROGRAMACIÓN EXTREMA	35
<u>7</u>	<u>PLANTEAMIENTO METODOLÓGICO.....</u>	<u>35</u>
7.1	FASES DEL PROYECTO	36
7.1.1	FASE 1. CONTEXTUALIZACIÓN DE LAS MEJORES PRÁCTICAS PARA EL DESARROLLO DE SOFTWARE.....	36
7.1.2	FASE 2. ESTABLECER UN CONJUNTO DE MEJORES PRÁCTICAS DE DESARROLLO DE SOFTWARE PARA LOS ESTUDIANTES DE MEDIA TÉCNICA	37
7.1.3	FASE 3. DISEÑAR UN INSTRUMENTO QUE PERMITA LA IMPLEMENTACIÓN DE LAS MEJORES PRÁCTICAS DE DESARROLLO DE SOFTWARE EN EL PROCESO FORMATIVO DE LOS ESTUDIANTES DE MEDIA TÉCNICA	37

7.2	FUENTES DE INFORMACIÓN Y ANÁLISIS	37
7.2.1	TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN.....	38
7.2.2	ANÁLISIS DE DATOS	39
8	<u>DESARROLLO METODOLÓGICO</u>	<u>39</u>
8.1	FASE 1. CONTEXTUALIZACIÓN DE LAS MEJORES PRÁCTICAS PARA EL DESARROLLO DE SOFTWARE	39
8.1.1	IDENTIFICACIÓN DE ACTORES Y ROLES	39
8.1.2	DISEÑO Y APLICACIÓN DE INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN	42
8.1.3	COMPONENTE ORGANIZACIONAL	52
8.1.4	COMPONENTE TALENTO HUMANO	58
8.1.5	COMPONENTE PROCESO DE DESARROLLO DE SOFTWARE	70
8.2	FASE 2: MEJORES PRÁCTICAS DE CALIDAD DE DESARROLLO DE SOFTWARE PARA LA MEDIA TÉCNICA EN PROGRAMACIÓN	74
8.3	FASE 3. PROPUESTA PARA LA IMPLEMENTACIÓN DE MEJORES PRÁCTICAS DE DESARROLLO DE SOFTWARE EN EL PROCESO FORMATIVO DE LOS ESTUDIANTES DE MEDIA TÉCNICA.....	79
8.3.1	INSTRUMENTOS PARA LA IMPLEMENTACIÓN DE MEJORES PRÁCTICAS DE DESARROLLO DE SOFTWARE EN LA MEDIA TÉCNICA EN PROGRAMACIÓN.	80
9	<u>CONCLUSIONES</u>	<u>89</u>
10	<u>TRABAJOS FUTUROS</u>	<u>92</u>
11	<u>BIBLIOGRAFIA</u>	<u>93</u>
12	<u>ANEXO: FORMULARIO - ENCUESTA A EMPRESARIOS SOBRE MEJORES PRÁCTICAS.....</u>	<u>103</u>

TABLA DE ILUSTRACIONES

Ilustración 1 Mercado mundial de servicios TI.....	1
Ilustración 2 Informe del proceso de evaluación de impacto.....	9
Ilustración 3 Nivel CMMI implementado por las empresas analizadas.....	54
Ilustración 4 Certificaciones de calidad obtenida por las empresas analizadas...	56
Ilustración 5 Nivel de desarrollo: Trabajo en equipo	60
Ilustración 6 Nivel de desarrollo: Orientación al logro	60
Ilustración 7 Nivel de desarrollo: Manejo de TIC	61
Ilustración 8 Nivel de desarrollo: Programar en un lenguaje	61
Ilustración 9 Nivel de desarrollo: Investigación y creatividad	62
Ilustración 10 Nivel de desarrollo: Comunicación Oral y escrita	62
Ilustración 11 Nivel de desarrollo: Pensamiento Algorítmico	63
Ilustración 12 Nivel de desarrollo: Pensamiento analítico-sistémico.....	64
Ilustración 13 Nivel de desarrollo: Documentación de un software	64
Ilustración 14 Nivel de desarrollo: Comprensión Organizacional.....	65
Ilustración 15 Nivel de desarrollo: Construcción de modelado de datos	65
Ilustración 16 Perfil profesional de desempeño en las fases de requisitos, análisis y diseño	68
Ilustración 17 Perfil profesional de desempeño en Fase de Desarrollo de Software.....	68
Ilustración 18 Perfil profesional de desempeño en prácticas organizacionales...	69

TABLA DE CUADROS

Cuadro 1	Características modelo de calidad de software	23
Cuadro 2	Características y sub - características de la norma ISO 9126-1.....	31
Cuadro 3	Características empresas analizadas	42
Cuadro 4	Relación proceso - producto software	43
Cuadro 5a	Relación prácticas organizacionales y su fuente	45
Cuadro 5b	Relación prácticas organizacionales y su fuente	46
Cuadro 6	Relación prácticas requisitos y fuente	47
Cuadro 7	Relación prácticas Análisis – Diseño y fuente	48
Cuadro 8a	Relación prácticas Desarrollo y fuente	49
Cuadro 8b	Relación prácticas Desarrollo y fuente	50
Cuadro 9	Variables a evaluar.....	52
Cuadro 10	Certificaciones obtenidas por las empresas analizadas.....	53
Cuadro 11	Selección de prácticas Organizacionales.....	76
Cuadro 12	Selección prácticas de requisitos	77
Cuadro 13	Selección prácticas análisis – diseño.....	78
Cuadro 14	Selección prácticas desarrollo.....	79
Cuadro 15a	Instrumento verificación prácticas organizacionales	83
Cuadro 15b	Instrumento verificación prácticas organizacionales	84
Cuadro 15c	Instrumento verificación prácticas organizacionales	85
Cuadro 16	Instrumento verificación prácticas requisitos	86
Cuadro 17	Instrumento verificación prácticas análisis - diseño	87
Cuadro 18a	Instrumento verificación prácticas de Desarrollo	88
Cuadro 18b	Instrumento verificación prácticas de Desarrollo.....	89

RESUMEN

Este proyecto centró la atención en la identificación de buenas prácticas de calidad para el desarrollo de software, centradas en el proceso y no en el producto de software, viables de ser incorporadas en la formación de los estudiantes de la media técnica en informática y programación en instituciones de educación básica y media pertenecientes a la Alianza Futuro Digital Medellín; esto con el fin de aportar elementos significativos en la formación técnica, que se verán reflejados en la calidad del producto software.

Para ello, se validaron las prácticas más reconocidas en el sector, respecto a su implementación cotidiana en las empresas de software. En esta actividad participaron empresas de software de la ciudad de Medellín, vinculadas a la Alianza Futuro Digital.

Como resultado de este proceso, se identificaron aquellas prácticas más frecuentes en su aplicación, y se llevaron a un instrumento para sugerir su incorporación didáctica y metodológica en los currículos de la media técnica en informática y programación de las instituciones de educación básica y media pertenecientes a la Alianza Futuro Digital Medellín.

Palabras claves: Calidad de software, Buenas Prácticas, Desarrollo de Software.
Media técnica.

1 INTRODUCCIÓN

La industria del software crece día a día, siendo la India el principal vendedor con un 70% de penetración en el mercado, y siendo 9 de las 10 primeras empresas líderes en el sector del software, norteamericanas. De ahí, que gran parte del talento humano indio vaya a estas empresas, ya que tienen excelentes competencias como programadores y además dominan el inglés. (Conferencia de las Naciones Unidas sobre comercio y desarrollo, 2012)

En la consolidada industria del software se destaca hoy China, Francia, Alemania, India, Reino Unido y Estados Unidos tal como aparece en la ilustración 1.

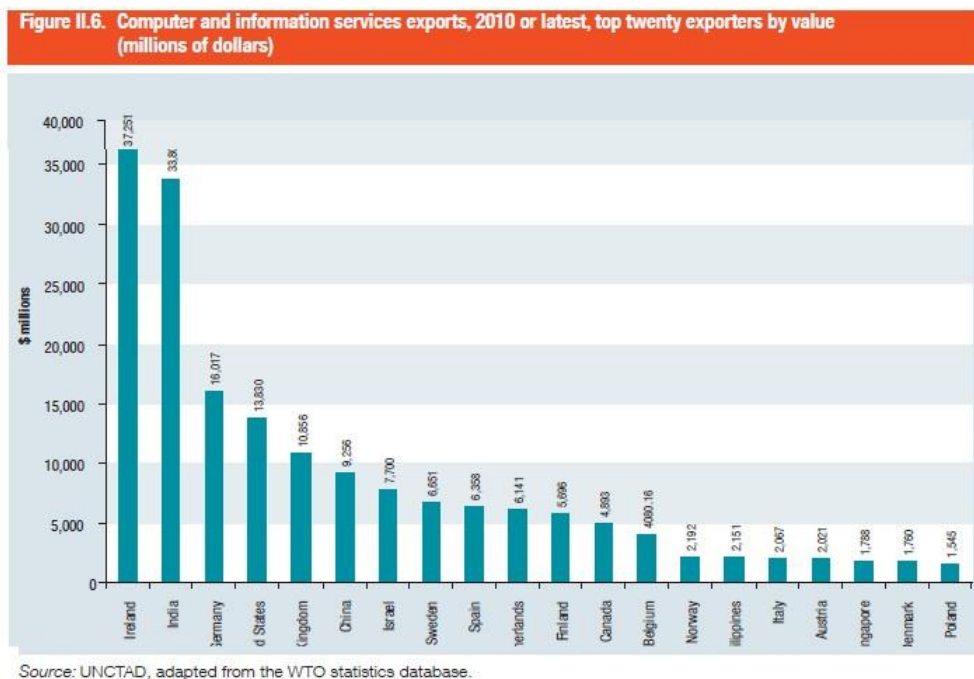


Ilustración1: Mercado mundial de servicios TI

Fuente: UNCTAD basado en información del Everest Research Institute

El predominio de la India en el mercado mundial de servicios de Tecnología e información (TI), es evidente con el 59% del total, mientras que a nivel de Latinoamérica la participación es mínima, sólo figura México con un 1%.

El proceso de la India e Irlanda se ha dado gracias a políticas de gobierno que han invertido en el talento humano tanto para la formación de competencias específicas del sector, como para aquellas de soporte (derecho, gerencia, mercado).

A esto se suma potenciar el bilingüismo, la oferta formativa relacionada con el sector productivo en proyectos de investigación e innovación, representados además en la creación de parques tecnológicos integrados a la industria para poder atraer la inversión extranjera y competir a escala con otros países.

Teniendo en cuenta este contexto global, desde los informes sobre los cargos más demandados en 2013 y el exponencial crecimiento de la industria informática, hasta las competencias requeridas a la hora de contratar nuevos empleados, el panorama actual no es muy diferente al de Colombia.

De acuerdo con Fedesoft, la escalabilidad en el mercado externo de software ha sido posible no sólo por la especialización en “productos y servicios de bajo costo, como ha sucedido en los últimos años sino porque en un futuro cercano la industria de software en los países en desarrollo debe focalizar su atención hacia actividades intensivas en conocimiento y, por tanto, de mayor valor agregado” (Fedesoft, 2012).

De esta manera se abre una gran posibilidad para Colombia a partir de los avances que ha logrado la Alianza Futuro Digital.

2 PLANTEAMIENTO DEL PROYECTO

En la actualidad, en el panorama de las Tecnologías de la Información y la Comunicación - TIC en especial el desarrollo de software, surgen dos grandes interrogantes relacionados con ¿Cómo desarrollar software? Y ¿Cómo medir la calidad del software desarrollado?, de tal forma que cumpla con las características de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad; que determinarán su usabilidad y existencia en el tiempo. Esto implica la utilización de metodologías o procedimientos estándares para desarrollar cada una de las etapas o fases del ciclo de vida del software, sin dejar de lado cada una de las personas que intervienen dentro del proceso y que juegan un papel trascendental en toda esta situación.

Hoy se requieren desarrollos más rápidos y que cumplan con las necesidades del cliente, superar los desfases de tiempos y costos de antes, dicha exigencia del medio hace que deba revisarse como el equipo del proyecto software pueda alinearse de manera que se logren mejores resultados.

Los equipos de desarrollo están cada día aprendiendo a mejorar su desempeño integrando metodologías cada vez más eficaces para esos fines, el fruto de esta investigación permitirá llevar al aula estrategias didácticas que permitan desarrollar buenas prácticas de la ingeniería de software al alcance de los estudiantes de la media técnica en programación de la Alianza Futuro Digital , de esta manera se podrá garantizar que el mercado laboral recibirá talento humano que desde su formación inicial posee buenas prácticas que permitirán eficiencia en procesos individuales, grupales y organizacionales.

El reto se centra en como combinar con precisión los elementos de la ingeniería de software en el aprendizaje pero logrando motivación, comprensión y aprendizaje por parte de los estudiantes.

3 DESCRIPCIÓN DEL PROBLEMA

El software lo encontramos en todas partes, en estos momentos existen programación para gran cantidad de actividades con el fin de hacer nuestras vidas más confortables, eficientes y efectivas. Por esta razón la ingeniería de software y los procesos de calidad son más importantes que nunca, las buenas prácticas de la ingeniería del software deben asegurar que los procesos que estos realizan sean eficientes y eficaces, pues gran parte de nuestras vidas depende de ellas.

El software permea hoy todos los ámbitos humanos, facilita la vida, cumple unas especificaciones muy precisas, resolviendo problemas cotidianos; de ahí que la apuesta que debe hacerse es según Fedesoft “Federación Colombiana de la Industria del Software y Tecnologías Informáticas Relacionadas” (Fedesoft, 1999) es todo un potencial, enfocado a Desarrollo a la Medida para dar respuesta a las necesidades de los clientes, desarrollo de aplicaciones WEB, computación en la nube, seguridad informática, tecnologías transversales o de propósito general que corresponden a sectores intensivos en conocimiento con múltiples campos de aplicación industrial y planeación de sistemas, SOA e integración: mediante la Arquitectura Orientada a Servicios (SOA) .

A nivel educativo, el Gobierno Nacional de Colombia a través del documento Conpes 3360 de junio de 2005 realizó un diagnóstico sobre el estado de la educación técnica y tecnológica del país, el cual arrojó la necesidad de implementar un proyecto para fortalecerla en cuanto a “la cobertura, calidad y pertinencia, acorde con las necesidades del sector productivo, el desarrollo

nacional y regional, y el avance de la ciencia y la tecnología” (Conpes, 2005), a cargo del Ministerio de Educación Nacional- MEN.

A nivel local y regional, dentro de las políticas de la ciudad de Medellín y el departamento de Antioquia desde 2004, se reconoce a la “industria del software como un sector transversal y estratégico para el desarrollo de la ciudad y el departamento, un sector que provee servicios y productos de alto valor agregado, que no solamente contribuyen a la competitividad de las industrias, sino que también tiene un alto potencial de exportación y generación de empleos que requieren personal calificado”(Agudelo,2013). Esto ha sido posible a la formación en las universidades, el compromiso del sector productivo y a la visión de futuro de convertir la ciudad de Medellín en Silicon Valley de Latinoamérica.

Por lo anterior, las iniciativas de gobierno apuntan “hacia una ciudad con una amplia y eficiente conectividad e integración con el mundo, por medio de sistemas tecnológicos modernos, uso y apropiación de las TIC para la inclusión digital, la interacción ciudadana y la generación e intercambio de conocimiento” (Agudelo, 2013). El sector software tiene entonces excelentes posibilidades en el campo de exportación y un gran posicionamiento a nivel nacional.

Estas son apenas algunas muestras del crecimiento tecnológico que se vive en Colombia y que es retomado por iniciativas locales como Ruta N, plan CTI, Intersoftware, Alianza Futuro Digital Medellín, Cluster TIC y las universidades que forman para el sector, tratando de enfrentar dificultades como: encontrar personal especializado que pueda trabajar para suplir la demanda de productos que se requieren, teniendo en cuenta que los egresados de TI en Colombia requieren entrenamiento adicional para poder cumplir los requisitos que exige la industria; ya que únicamente el 14% del total de los egresados se encuentra listo para ser productivo desde su contratación, solo el 19% de egresados es bilingüe y existe una disminución en un 5% anual de los graduados de Ingeniería de sistemas e informática. (Agudelo, 2013).

En la ciudad de Medellín a partir del año 2006, comenzó la Alianza Futuro Digital como una apuesta de los sectores educativos (SENA, Universidad EAFIT, Politécnico Colombiano Jaime Isaza Cadavid, Tecnológico de Antioquia) productivo (Créame e Intersoftware) y de gobierno (Secretaría de Educación de Medellín) para posicionar la industria del software de Medellín en mercados internacionales y favorecer la pertinencia de la articulación entre la Educación Media y la Educación Superior, por ciclos propedéuticos.

Actualmente, la Alianza Futuro Digital Medellín es un referente en la ciudad para la estrategia “Medellín, Ciudad Clúster” lo que significa que el clúster TIC está apoyándola para que continúe su crecimiento. Se une además, el “Fondo de Ciencia, Tecnología e Innovación, enmarcado en el Plan de Ciencia y Tecnología de Medellín, apoyando a Ruta N con la financiación de programas, proyectos y actividades con alto contenido de ciencia y tecnología, orientadas al fortalecimiento de las capacidades de innovación de los clúster estratégicos de la ciudad”. (Taborda, 2012).

Una de las mayores dificultades que encuentran los empresarios al momento de contratar un programador, es la falta de conocimientos en procesos de calidad y aunque en el fondo todos conocen los procesos que se deben realizar, estos no son aplicados al momento de estar trabajando en un proyecto de desarrollo de software. Esta dificultad la deben asumir los empresarios que deben cualificar al personal con la desventaja que algunos empleados cuando son certificados se cambian rápidamente a otras empresas que les ofrezcan mayor salario.

También se debe tener en cuenta que algunas empresas en el mercado no utilizan procedimientos, mejores prácticas o lineamientos de procesos de calidad en el desarrollo de software; y son pocas las que están certificadas y cuentan con un buen nivel de madurez en los procesos.

Todo esto lleva a que un producto software salga al mercado con deficiencias lo cual influye directamente en la calidad del producto, luego su mantenimiento postventa será muy alto, pues se deben realizar reprogramaciones que enfrentan a las empresas a situaciones críticas como son el aspecto financiero, seguridad, credibilidad, etc.

Por su parte, el Ministerio de Educación Nacional a través de la Ley 749 de 2002 introduce en el sistema educativo colombiano la formación por ciclos con carácter propedéutico¹, específicamente en las áreas de ingenierías, tecnología de la información y la administración.

Posteriormente, la Ley 1188 de 2008, la cual regula el registro calificado de programas de Educación Superior, amplía la posibilidad de formación por ciclos a todas las áreas del conocimiento. Según esta ley: "Todas las instituciones de Educación Superior podrán ofrecer programas académicos por ciclos propedéuticos hasta el nivel profesional en todos los campos y áreas del conocimiento dando cumplimiento a las condiciones de calidad previstas en la presente ley y ajustando las mismas a los diferentes niveles, modalidades y metodologías educativas" (Ministerio de Educación Nacional, 2008).

Esto ha permitido que en Medellín, se baje la formación técnica a la educación media, iniciando con 16 Instituciones de educación básica para el año 2.006 y 36 instituciones educativas para 2015 vinculadas a la Alianza Futuro Digital. Esta alternativa permite a los estudiantes que cursan las medias técnicas en informática y programación avanzar en su formación con el Politécnico Jaime Isaza Cadavid, institución universitaria de la ciudad de Medellín, que adecuó sus currículos de nivel técnico y tecnológico con el fin de favorecer al estudiante de la media técnica para la obtención de títulos por ciclos propedéuticos que le permitan acceder al mundo laboral.

¹ "Los ciclos son unidades interdependientes, complementarias y secuenciales; mientras que el componente propedéutico hace referencia al proceso por el cual se prepara a una persona para continuar en el proceso de formación a lo largo de la vida, en este caso particular, en el pregrado."(Mineduccion.gov.co).

La media técnica, especialidad Informática – Programación, desarrollada en el área metropolitana se encadena con la tecnología en Sistematización de datos, lo que permite que estudiantes de grado 10° y 11° con edades comprendidas entre 14 y 17 años y estrato socioeconómico 1, 2 y 3 puedan tener un reconocimiento de un año de estudio en las Instituciones Universitarias para continuar, una vez el estudiante se gradúe del colegio, en dichas instituciones estudiando un año más para obtener titulación como Técnico Profesional en Sistematización de Datos, luego en la Tecnología y hasta la Ingeniería de Sistemas. Aunque el estudiante debe tener conocimiento acerca de la gestión de un proyecto de desarrollo software, su salida ocupacional está enfocada en la programación lo que hace que el énfasis se encuentre allí.

La Alianza Futuro Digital le apuesta al desarrollo de la ciudad y ha sido soporte para un proceso a largo plazo que impulsa el desarrollo de la región pero que trasciende lo nacional e internacional, pensando en varios frentes: El talento humano, los currículos, la infraestructura, soporte al talento humano, entre otros. Ver Ilustración 2.

- **Talento Humano**

Este punto es fundamental porque atiende la formación de los estudiantes, determina unas competencias mínimas a lograr en la formación de la media y otras para la formación técnica y tecnológica.

De acuerdo al estudio de la Alianza Futuro Digital Medellín en el 2012, el perfil del estudiante de la media técnica está dado por:

- Programar en un ambiente específico a partir de un diseño.
- Brindar soporte básico de aplicativos

- Operar sistemas.



Ilustración 2: Informe del proceso de evaluación de impacto.

*Fuente Alianza Futuro Digital
Medellín, 2010*

Y el perfil del Técnico profesional es:

- Proveer soluciones de software bajo estándares de programación
- Analizar , diseñar, probar e implementar aplicaciones
- Realizar documentación técnica del sistema.
- Administrar bases de datos y sistemas de información
- Coordinar la atención al usuario final

Brindar soporte al usuario final.

El talento humano docente está conformado por docentes articulados² quienes pertenecen a la Institución Educativa de educación media, los docentes articuladores³ que vienen de las Instituciones de Educación Superior, y juntos imparten los módulos durante un número de horas semanales que oscilan entre 7 y 12 horas. Hay otros actores como son los asesores de proyecto quienes además son gerentes de éste. Además estos proyectos que desarrollan los estudiantes de la Institución educativa durante estos dos años (10° y 11°), pueden continuarlos cuando avancen en la formación técnica profesional en programación de sistemas de información.

En el caso de las Instituciones Educativas que se encuentran articuladas con el SENA, los estudiantes obtienen su certificación como técnico laboral y el proceso es desarrollado por un instructor Sena con una intensidad de 4 horas a la semana y un docente de la Institución educativa que imparte otras 7 horas.

Currículo

En la definición de los módulos de aprendizaje y el enfoque del Proyecto Pedagógico Integrador (PPI), los estudiantes trabajan en un sin número de módulos de los cuales deben aplicar conceptos en la realización de un Proyecto Pedagógico Integrador (PPI), el cual consiste en resolver un problema a partir del diseño y creación de una aplicación. Las competencias fundamentales que deben desarrollar los estudiantes giran alrededor de la programación, el conocimiento y uso de diversos lenguajes, la práctica, los errores de sintaxis, de lógica, la interpretación de diagramas, de código y la reutilización de código, entre otros.

El Proyecto Pedagógico Integrador (PPI) debe evidenciar las competencias que desarrollan en cada uno de los módulos vistos en cada grado. Este proyecto

² Docente articulado es el docente que labora en las instituciones de Educación básica y media, acompañan a los estudiantes en su día a día de formación.

³ Docente articulador es el orientador del programa desde las instituciones universitarias.

tendría una duración de dos años y que podría tener continuidad en la técnica profesional. “El objetivo del PPI es aplicar de manera integral las competencias desarrolladas en los diversos módulos de los niveles en proceso ascendente de lo simple a complejo, con proyecciones hacia la creatividad y el inicio de actividades investigativas” (Alianza Futuro Digital Medellín, 2012) se constituye entonces en una herramienta de auto aprendizaje, donde:

- Los proyectos son desarrollados en grupos de 3 o 4 personas, el cual es de libre escogencia; después de haber dado una inducción acerca del trabajo en equipo y los roles que deben desempeñar en la ejecución del proyecto.
- El proyecto debe dar solución a una necesidad de manejo de información para una pequeña o mediana empresa del sector, bien sea un supermercado, una tienda, una videotienda, la biblioteca institucional, un centro médico, una panadería, entre otras.
- El software debe agilizar los procesos dentro de la organización, mejorar la búsqueda y almacenamiento de la información y por ende dar un valor agregado al servicio que ofrece la pequeña o mediana empresa.
- Para desarrollar este proyecto los estudiantes iniciarán las fases del ciclo de vida del software en grado décimo con la elaboración del prototipo navegaciones del software y en grado undécimo desarrollarán la conexión a la base de datos y programarán los métodos necesarios en cada uno de las Cuadros maestras y las Cuadros transaccionales, así como la elaboración de informes, nociones en pruebas e implantación.
- Para desarrollar el proyecto los estudiantes deben seguir el ciclo de vida del software y realizar una serie de entregables con fechas establecidas, estándares de presentación y sustentaciones del mismo.
- Cada estudiante debe llevar un control de tiempos y registro de errores pero no tenían disciplina para ello por lo que lo copiaban para todos los miembros del grupo como si fuera igual. En la actualidad se hace un seguimiento al cronograma del proyecto.
- Los mejores proyectos institucionales pueden asistir a diferentes eventos a

nivel local y/o regional.

Infraestructura

Para este proceso es fundamental la infraestructura por eso las Instituciones educativas que tienen este programa están dotados de salas de sistemas, software necesario y recursos para desarrollar los procesos de enseñanza aprendizaje establecido por cada una de las Instituciones Articuladoras. Por otra parte se cuenta con el Vivero del software, como un símil de una fábrica de software, espacio de construcción colectiva entre estudiantes de la educación media, educación superior y los empresarios.

El Vivero del software es un ambiente para el desarrollo y fortalecimiento de la formación del talento humano en cuanto a las competencias específicas de los futuros técnicos profesionales y tecnólogos desarrolladores de software que atiende las demandas del mundo laboral regional, incorporando estándares de internacionalización. Esto implica la formación docente para generar conocimientos y nuevas prácticas educativas que permitan la formación de los ciudadanos del siglo XXI (Laboratorio de las TIC).

Soporte al talento humano

El proceso de media técnica y el desarrollo del Proyecto Pedagógico Integrador (PPI), ha estado acompañado desde el 2009 por el programa Mentores empresariales en el que participan empresas aliadas, desarrolladoras de software y afines, que brindan a los jóvenes charlas motivacionales, asesorías a sus proyectos y visitas a sus instalaciones con el fin de acercarlos al ambiente laboral.

En algunos casos, estas empresas contratan a los jóvenes que de alguna manera son referenciados por los docentes de las instituciones educativas, luego son citados a entrevista y presentan exámenes para cumplir el proceso de selección.

Todos esperamos calidad en los productos que se adquieren, en los servicios que se solicitan, por eso parece obvio hablar de calidad pero al interior de las organizaciones, no es así, se requiere de una estructura organizacional pensada estratégicamente que incremente el índice de éxito en el proceso.

De acuerdo al informe de la Alianza Futuro Digital en el 2012 y entrevistas a los empresarios de la industria del software, se considera que los proyectos que estos jóvenes de la media técnica desarrollan no tienen la calidad adecuada y que ninguna empresa a la que se le desarrolla va a implementarla. Se está desconociendo que son jóvenes de 14 años que se encuentran desarrollando aún sus competencias básicas y que están construyendo sus bases, pero que indudablemente deben continuar el proceso de formación. Por otro lado, al tener un perfil de salida de la media técnica en programación, pueden desempeñarse únicamente en codificación, teniendo que demostrar habilidades para ocupar otros roles. Además afirman que los actuales profesionales vinculados a la industria tienen “dificultades en competencias básicas y específicas tales como bilingüismo, calidad en el desarrollo de software, metodologías de testing, seguridad informática, procesos de comercialización de tecnología, entre otros temas que son diferenciadores para las empresas del sector”. (Alianza Futuro Digital Medellín, 2012).

De acuerdo a entrevistas con John Jairo Monsalve coordinador académico del Politécnico Jaime Isaza Cadavid el tema del talento humano docente es complejo en la medida que algunos no tienen la experiencia laboral dentro del mundo de la empresa, desconociendo la realidad y otros docentes conocen este campo pero no tienen la pedagogía para llevar la ingeniería de software al aula de clase.

A la Alianza Futuro Digital le interesa impactar con buenas prácticas incorporadas en la formación de los estudiantes pues esto garantiza a futuro que pueda entregarse un buen producto software.

Los proyectos académicos de desarrollo de software deben proporcionar a los estudiantes la oportunidad de poner en práctica el conocimiento y las habilidades adquiridas en cursos anteriores, en contextos de problemas de diferente alcance, desde la concepción hasta la implementación de la solución. Los escenarios para estas experiencias deben emular la realidad de la industria con la distribución de roles, los compromisos con el tiempo, la calidad, y la resolución de imprevistos, todo esto tan cerca como sea posible, utilizando clientes reales (docentes, líderes de investigación, directores de proyecto), procesos, herramientas y criterios de tiempo y calidad.

Por su parte al bajar la formación técnica al ambiente escolar, implica que los estudiantes de la media técnica en programación sean jóvenes que necesitan invertir un buen número de horas en su formación y cursar varios módulos. Esta formación técnica que logran los estudiantes al terminar el colegio es una parte básica, que ellos deben continuarla en las Instituciones de Educación Superior, por lo que es preciso garantizar que estos jóvenes continúen con entusiasmo y sobresaliendo en este campo que tanta demanda tiene.

De acuerdo con (Pressman, 2002) la ingeniería de software se caracteriza por ser una tecnología multicapa, en la cual la capa más interna corresponde al enfoque de la calidad como centro fundamental del proceso de desarrollo, luego viene el modelo de proceso como un marco de trabajo, se unen luego los métodos como la forma de hacer las cosas y por último las herramientas. Si los jóvenes desde su formación no comprenden la importancia de dicho proceso será complejo lograr resultados óptimos, de ahí las actuales falencias encontradas de acuerdo al proceso que se lleva.

En la media técnica los estudiantes desarrollan un Proyecto Pedagógico Integrador (PPI) en grupos de 3 o 4 personas, desarrollando módulos de sistemas de información que pretenden soluciones a problemas de manejo de información de

microempresa o empresas del sector como tiendas, minimercados, panaderías, carnicerías, entre otras. Este proyecto debe evidenciar las competencias trabajadas en los diferentes módulos y seguir las fases del ciclo de vida del software, realizando una serie de entregables en cada una de estas fases. Además los estudiantes deben construir sus propios cronogramas de actividades para planear el desarrollo de sus proyectos y llevar un control de tiempo invertido en cada una de las actividades planeadas de forma individual.

Todo el proceso es desarrollado y apoyado por diferentes entes: docente articulado y articulador, asesor de PPI y docentes de las áreas transversales: matemáticas, humanidades: lengua castellana e idioma extranjero, artística, tecnología e informática y ética.

En el desarrollo de los procesos de enseñanza aprendizaje de cada uno de los módulos y su aplicación en el Proyecto Pedagógico Integrador (PPI) se evidencian algunos puntos como:

- Las instituciones educativas cuentan con recursos humanos y tecnológicos para desarrollar los procesos de enseñanza aprendizaje, sin embargo algunos de los estudiantes no poseen este tipo de herramientas en sus hogares que les permita fortalecer estos procesos.
- La calidad del software está dada por los métodos, herramientas, y procedimientos propios de la ingeniería de software; y en el proceso de enseñanza aprendizaje su correcta aplicación depende en gran medida de la competencia que posean los actores que intervienen en el proceso (Docente articulado y articulador).
- Los estudiantes no logran realizar estimaciones de tiempo adecuadas que les permita proyectar el desarrollo de su solución a través del tiempo.
- Los esquemas de actividades que almacenan los tiempos empleados en la ejecución de las actividades planeadas no reflejan tiempos reales ya que ellos solo diligencian el esquema al momento que debe ser entregado como

“tarea”, y no a conciencia como parte del proyecto.

- El trabajo en equipo es reducido, lo que se ve reflejado en la sobrecarga de trabajo en uno (s) de los miembros del equipo, mientras que otros se dedican a hacer actividades de documentación, pero no de programación.
- En el desarrollo del PPI se requiere establecer procesos de comunicación eficientes entre los miembros de los equipos y de los mismos actores involucrados (Docentes y asesor de PPI), sin embargo es una de las grandes dificultades de los estudiantes y de los actores que dirigen el proceso.
- El desarrollo del Proyecto Pedagógico Integrador (PPI), es un proceso evolutivo, que evidencia su avance en los entregables que deben desarrollar los estudiantes y en las versiones de estos, sin embargo en varias ocasiones los estudiantes demuestran no controlar las versiones de sus proyectos, entregables y avances, lo que genera tiempo y esfuerzo adicional para la búsqueda de versiones a entregar.
- La calidad del software en gran medida radica en la comprensión del problema y los requisitos funcionales para el software, en este sentido los estudiantes carecen de una comprensión global del problema y una comprensión exacta de los requisitos por lo cual deben realizar innumerables entrevistas e indagación con el cliente.
- Se conocen los distintos roles para el desarrollo de software (analista, diseñador, desarrollador o programador y tester) para el desarrollo de software pero se genera en ocasiones apatía por el programador al pensar que es una labor difícil.
- Se tiene la creencia que una vez se escribe el código y este funciona, el trabajo ha terminado.
- Pareciera que sólo hasta que el programa no se ejecute en contexto real no hay forma de comprobar su calidad.
- Los estudiantes que ingresan a media técnica no poseen buenas bases en la solución de problemas cotidianos, que emplea los recursos disponibles.
- En el proceso de media técnica los estudiantes que se encuentran articulados

con el Politécnico Colombiano Jaime Isaza Cadavid ven en su proceso de formación, en total nueve módulos del técnico profesional organizados en un cronograma establecido por dicha institución, sin embargo los espacios para la práctica son reducidos por lo que en algunas ocasiones se debe emplear más tiempo en la práctica y reducir parte de la teoría.

- En procesos de diseño de pruebas y ejecución de pruebas son escasos y son desarrollados al final del desarrollo del proyecto.

En resumen, el problema radica en lograr que los estudiantes en formación técnica, comprendan el ciclo del desarrollo de software, como un sistema que articula etapas de desarrollo del producto, y por tanto que la aplicación de técnicas, metodologías y buenas prácticas les asegura que su aporte al producto software sea de calidad.

Esto lleva a la pregunta: ¿Cómo implementar mejores prácticas de calidad para el desarrollo de software en el proceso de formación en la media técnica en programación, en las Instituciones educativas pertenecientes a la Alianza Futuro Digital?

4 ESTADO DEL ARTE (ANTECEDENTES)

Poder medir la calidad de un producto software y de un proceso de desarrollo ha representado una constante búsqueda de uso de metodologías y estrategias que permitan controlar muchos factores, de ahí que las experiencias encontradas no los solucionan todos, más bien será la mezcla de varias las que permita mejorar dichos procesos.

Una de las experiencias donde se comprueba que la calidad del proceso de desarrollo influye directamente en la calidad del producto es la integración del modelo de Calidad del Producto (Ortega et al., 2001) y el modelo de Calidad del Proceso de Desarrollo (Pérez et al., 2001), ambos desarrollados en el Laboratorio

de Investigación en Sistemas de Información (LISI) de la Universidad Simón Bolívar de Venezuela, en un prototipo de Modelo Sistémico de Calidad (MOSCA) para la Medición de la Calidad de los Sistemas software, soportado por los conceptos de la Calidad Total Sistémica.

Este modelo fue validado entre los años 2002 y 2004 en Venezuela, y arroja el nivel de calidad que puede variar entre Nulo, Básico, Intermedio y Avanzado, especificando los procesos que se deben mejorar en la empresa y las características que no son satisfechas por el producto de software desarrollado. De esta manera, se combinan los criterios relacionados con el producto y con el proceso.

Luego de haber evaluado MOSCA a través de la aplicación del método Análisis de Características por Caso de Estudio (Kitchenham, 1997) en dos (2) oportunidades, se llegó a la conclusión que el modelo es una herramienta efectiva de análisis y medición de la calidad sistémica global, pues se analizan aspectos del producto, del proceso y su relación con el medio ambiente. Las empresas evaluadas mostraron satisfacción cuando se aplicó el modelo y sobre todo cuando se les presentaron los resultados obtenidos, sus respectivas explicaciones y las acciones que debían emprender para mejorar su Calidad Sistémica.

Para futuras investigaciones se recomendó además de la aplicación de cuestionarios directos e indirectos, instanciar MOSCA para diferentes tipos de Sistemas software y la creación de un método de medición de la calidad basado en MOSCA. (Mendoza, Pérez, 2005)

Una segunda experiencia se desarrolla en el Programa de Ingeniería de Sistemas y Computación de la Universidad del Quindío en Armenia (Colombia) en el año 2011, su objetivo era incorporar buenas prácticas y conceptos de calidad en el desarrollo de software desde etapas tempranas en la formación de los estudiantes de pregrado. Se intervino el currículo y se crearon ambientes para el desarrollo de la propuesta.

La apropiación de las prácticas de Personal Software Process (PSP) y Team Software Process (TSP) durante el proceso de formación básica, permite a los estudiantes comprender el concepto de proceso, la adquisición de hábitos relacionados con el registro de información y posteriores mediciones. De igual forma, la recolección de datos individuales y de equipo permite tener una línea de referencia para la planificación y el desarrollo de proyectos futuros.

Se han identificado tres factores principales que influyen en la enseñanza de PSP: el entorno de trabajo, el nivel de cobertura, y las herramientas de apoyo (Börstler, Carrington, & Hislop, 2002). Ahí es donde entra la importancia de una estrategia de aprendizaje para llegar a un buen resultado para incorporar conceptos de calidad durante el proceso de formación de los estudiantes de pregrado.

5 OBJETIVOS

5.1 Objetivo General

Desarrollar una propuesta para implementar buenas prácticas de calidad para desarrollo de software en los procesos de formación de la media técnica en las Instituciones Educativas pertenecientes a la Alianza Futuro Digital – Medellín, alineadas con las necesidades de las empresas del sector.

5.2 Objetivos Específicos

- Identificar un conjunto de mejores prácticas para el desarrollo de software, usadas en la industria del software por las empresas del grupo Intersoftware.
- Seleccionar de las mejores prácticas identificadas en las empresas, aquellas que aporten al desarrollo de las competencias que deben desarrollar los estudiantes durante su formación en la media técnica.
- Diseñar una estrategia que permita la implementación de las mejores

prácticas en el proceso formativo de los estudiantes de la media técnica.

6 MARCO DE REFERENCIA

6.1 Proceso de desarrollo de Software

La ingeniería de software es una tecnología multicapa en la que, según Pressman (2005), “se pueden identificar: los métodos, el proceso (que es el fundamento de la Ingeniería de Software, es la unión que mantiene juntas las capas de la tecnología) y las herramientas (soporte automático o semiautomático para el proceso y los métodos)”. De ahí la importancia antes de empezar a definir las más importantes metodologías y estándares de calidad, de aclarar que los modelos para el desarrollo de software son una representación abstracta o marcos de un proceso y que pueden ser adaptados para crear procesos más específicos, aquí encontramos los tradicionales cascada, evolutivo (espiral), basado en componentes, entre otros.

Según Silva (2001) “desde 1985 hasta el presente, han ido apareciendo herramientas, metodologías y tecnologías que se presentaban como la solución definitiva al problema de la planificación, previsión de costos y aseguramiento de la calidad en el desarrollo de software”. Donde Piattini (1996), llega a la definición de metodología de desarrollo como “un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevo software”. Las metodologías atienden tres necesidades principales:

- Mejores aplicaciones, conducentes a una mejor calidad.
- Un proceso de desarrollo controlado.
- Un proceso normalizado en una organización, no dependiente del personal.

Se pueden enumerar una serie de características (Piattini, 1996) que debe tener la metodología y que influirán en el entorno de desarrollo:

- Reglas predefinidas
- Determinación de los pasos del ciclo de vida
- Verificaciones en cada etapa
- Planificación y control
- Comunicación efectiva entre desarrolladores y usuarios.
- Flexibilidad: aplicación en un amplio espectro de casos
- De fácil comprensión
- Soporte de herramientas automatizadas.
- Que permita definir mediciones que indiquen mejoras
- Que permita modificaciones
- Que soporte reusabilidad del software.

El uso de las metodologías permiten identificar buenas prácticas que van a permitir desarrollar mejor el proceso y obtener mejores resultados, por ende obtener calidad.

6.2 Calidad en el proceso del software

Primero es importante entender cuál es el significado de software, “los programas de ordenador, los procedimientos y, posiblemente la documentación asociada y los datos relativos a la operación del sistema informático” (IEEE, 1990).

La calidad es un término que ha adquirido gran relevancia con el paso del tiempo, ya que es considerada como uno de los principales activos con los que cuenta un país para mejorar su posición competitiva global (Ivanisevich et al., 1997).

El estándar IEEE 6.10-1990 [IEEE, 1990] da la definición de calidad como “el grado con el que un sistema, componente o proceso cumple con los requisitos especificados y las necesidades o expectativas del cliente o usuario”.

Según Pressman (2002) la calidad del software es “la concordancia con los

requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”. La ausencia de defectos, la aptitud para el uso, la seguridad, la confiabilidad y la reunión de especificaciones son elementos que están involucrados en el concepto de calidad del software. Sin embargo, la calidad del software debe ser construida desde el comienzo, no es algo que puede ser añadido después (Humphrey, 1997).

(Cataldi, Lage, et. al.). Se refiere a que “El logro de la calidad puede tener tres orígenes. Calidad realizada, calidad programada y calidad necesaria. La primera es la que es capaz de obtener la persona que realiza el trabajo, la segunda es la que ha pretendido obtener y la tercera la que exige el cliente y que le gustaría recibir. La gestión de la calidad pretenderá que estas coincidan”. (p50).

Como lo plantea Abad (2012), en su artículo Framework para el desarrollo de software para entornos académicos:

La industria del software desde hace más de 20 años ha venido trabajando en una orientación por procesos, apoyada tanto por el estándar ISO 9000, como por modelos específicos para esta industria, como ISO/IEC 15504, ISO/IEC 12207, PSP (Personal Software Process), TSP (Team Software Process), Métrica 3, Moprosoft, CMM y posteriormente CMMi, entre muchos otros.

Como consecuencia, para las organizaciones que los han adoptado de forma disciplinada y consciente les ha significado una madurez en la ejecución y gestión del Proceso Software, que les permite realizar proyectos exitosos en términos de cumplimiento de tiempo, costo y alcance, calidad y satisfacción de los clientes. (Abad 2012, p 1.).

A continuación se presentan algunas características de los modelos de calidad iniciales que dieron origen a la Norma ISO. Ver cuadro 1:

Cuadro1 Características de modelos de calidad de software (Fuente: Construcción propia)

MODELO	CARACTERISTICAS
<p>Modelo de Calidad de McCall – 1977</p>	<p>Creado por Jim McCall para la Fuerza Aérea de los EE.UU. Buscando cerrar la brecha entre los usuarios y los desarrolladores.</p> <p>Identificó tres perspectivas principales para la caracterización de los atributos de calidad de un producto de software:</p> <ul style="list-style-type: none"> • Las operaciones del producto (características operativas básicas). • Revisión del producto (capacidad de cambiar). • Transición Producto (adaptabilidad a nuevos entornos). <p>(Scalone,2006)</p>
<p>De Boehm Modelo de Calidad – 1978</p>	<p>Creado por Barry W. Boehm</p> <p>Define un modelo jerárquico de las características de métricas de calidad de software (mediciones).</p> <p>Definió tres usos principales o requisitos básicos de software:</p> <ul style="list-style-type: none"> • La facilidad de uso, la fiabilidad y la eficiencia
<p>De Boehm Modelo de Calidad – 1978</p>	<ul style="list-style-type: none"> • Capacidad de mantenimiento, facilidad de identificación de lo que necesita ser cambiado, así como la facilidad de modificación y volver a probar. • Portabilidad, facilidad de cambiar el software para adaptarse a un nuevo entorno. <p>Identificó los siguientes factores de calidad: portabilidad, fiabilidad, eficiencia, capacidad de prueba, comprensibilidad y flexibilidad.</p> <p>(Scalone,2006)</p>

Existen varios puntos de vista frente a la definición de calidad y estándares, y el concepto de aseguramiento de la calidad del software.

Desde el punto de vista de la evidencia, el Institute of Electrical and Electronics Engineers (IEEE) la define como “Una guía planificada y sistemática de todas las

acciones necesarias para proveer la evidencia adecuada de que un producto cumple los requerimientos técnicos establecidos. Un conjunto de actividades diseñadas para evaluar el proceso por el cual un producto es desarrollado o construido.” (Vargas, 2009).

Por su parte el Software Engineering Institute (SEI) de Carnegie Mellon University desde el punto de vista de la visibilidad, dice “El aseguramiento de la calidad del software provee claro control del proceso que su parte siendo usado por el proyecto y del producto que se está construyendo.” (Vargas, 2009).

Desde el punto de vista del aseguramiento, Vargas referencia a Don Reifer cuando dice que “El aseguramiento de la calidad del software es el sistema de métodos y procedimientos usados para asegurar que el producto de software alcanza sus requerimientos. El sistema involucra la planificación, estimación y monitoreo de las actividades de desarrollo realizadas por otros” (Vargas, 2009, p 11)

El aseguramiento de la calidad es posible cuando se siguen métricas de calidad y se recogen medidas, que como estándares que son permiten cuantificar aspectos específicos de un proceso, de un producto o de un proyecto de ingeniería (Pressman, 2002).

Con datos de tamaño, tiempo y defectos, existen muchas formas de medir, evaluar, y manejar la calidad de un producto y para medir el proceso mediciones de calidad que brinda PSP y TSP que ayudan a los desarrolladores a medir su desempeño.

El SEI ha tomado la premisa de la gestión de procesos, “la calidad de un sistema o producto está muy influenciada por la calidad del proceso empleado para desarrollarlo y mantenerlo”. La adhesión a esta premisa se encuentra en los movimientos de calidad de todo el mundo, como lo muestra la International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) en su conjunto de estándares, es así como se tiene:

ISO 12207:2008 Procesos ciclo de vida

Esta norma desarrollada por el Institute of Electrical and Electronics Engineers (IEEE) y la Electronics Industry Association (EIA) establece un marco de referencia común para los procesos del ciclo de vida del software, contiene procesos, actividades y tareas para aplicar durante la adquisición de un sistema que contiene software y durante el suministro, desarrollo, operación y mantenimiento de productos o servicios software.

ISO 15504 Evaluación nivel de madurez

Esta norma es conocida como SPICE - *Software Process Improvement And Assurance Standards Capability Determination*.

Relaciona la organización de desarrollo software en la dimensión del proceso con los atributos del proceso en la dimensión de capacidad. Comprende la realización, planificación, definición, despliegue, medición e innovación de los procesos en función del nivel de madurez al que aspira la organización, que puede ser de 0 a 5. Se está convirtiendo en el estándar escogido por las empresas europeas para la evaluación de la capacidad de los procesos (nivel de madurez). ISO 15504 puede aplicarse en cualquier tipo de organización, modelo de ciclo de vida, metodología de desarrollo y de la tecnología utilizada (Martin, 2014).

En el desarrollo de software, al emplear paradigmas y metodologías, se va ganando una experiencia y una experticia que permite identificar prácticas que permiten hacer mejor los procesos que otras. De ahí que el concepto de *mejores prácticas para el desarrollo de software* se entienda como “el conjunto coherente de acciones que han rendido buen o incluso excelente servicio en un determinado contexto y que se espera que, en contextos similares, rindan similares resultados” (Dellacaná, 2008)

Es más lo que para una empresa desarrolladora, funcionó muy bien para otra no, ya que hay que tener en cuenta varias variables, como el talento humano, el clima

organizacional y el liderazgo que acompañan esa práctica.

De acuerdo a la Real academia española de la lengua, el concepto buenas /mejores prácticas puede ser equivalente a “mejores soluciones, mejores métodos, procedimientos más adecuados, prácticas recomendables, o similares.” (DRAE, 2014).

6.3 Modelos de referencia para la calidad en el proceso

La Ingeniería de Software ha evolucionado hacia modelos que permiten conocer e implementar la madurez de las organizaciones para desarrollar software como CMMi, SPICE, TSP, PSP; de igual forma se han generado grandes bases de conocimiento, como lo son RUP, SWEBoK, MSF, XP, OpenUP, entre otros, donde se consigna la mejor forma de construir y gestionar proyectos de software.

De acuerdo con las actuales investigaciones, hoy la tendencia de la industria de software nacional y mundial consideran 4 modelos que están impactando el proceso del software:

Proceso Unificado de Desarrollo de Software: El cual no es sólo una metodología de desarrollo de software adaptable y comprensible, basada en el desarrollo iterativo e incremental, sino que constituye una base de conocimiento de las mejores prácticas de la industria, realizado por los expertos de IBM, que se mantiene en constante evolución y mantenimiento.

Capability Maturity Model Integration DEV v.1.3 (CMMI): Constituye un modelo de referencia para la adopción de buenas prácticas de Ingeniería de Software en el área de construcción de software. El objetivo es mejorar la usabilidad de modelos de madurez, integrando varios modelos diferentes en un solo marco (framework). Fue creado por miembros de la industria, el gobierno y el SEI (Software Engineering

Institute). Entre los principales patrocinadores se incluyen la Oficina del Secretario de Defensa (OSD) y la National Defense Industrial Association.

Los modelos de CMMI son colecciones de las mejores prácticas que ayudan a las organizaciones para mejorar drásticamente la eficacia, eficiencia y calidad (Abad, 2012). El CMMI para desarrollo de software, un marco de mejora de procesos de software, analiza los distintos componentes de gestión de proyectos y recomienda un enfoque para la selección apropiada del ciclo de vida del proyecto, con el fin de tener calidad en los procesos y control de los productos, evaluar objetivamente los productos de trabajo, proporcionar objetivos propios, lograr objetivos específicos y métodos genéricos e institucionalizar un proceso gestionado y prácticas genéricas.

Project Management Body of Knowledge (PMBOK): El cual es el compendio de las mejores prácticas para la gestión de cualquier tipo de proyectos y cuenta con bastante aceptación por parte de la industria del software.

La Guía del PMBOK es un estándar en la administración de proyectos, aceptadas como las mejores prácticas y desarrollado por el Project Management Institute (PMI) en 1987, comprende dos grandes secciones; la primera sobre los procesos y contextos de un proyecto, la segunda sobre las áreas de conocimiento específico para la gestión de un proyecto. El PMBOK es un estándar reconocido internacionalmente (IEEE Std 1490–2003) que provee los fundamentos de la Gestión de proyectos que son aplicables a un amplio espectro de proyectos, entre los que se cuenta construcción, software, ingeniería, entre otros.

Metodologías ágiles: Basados en el manifiesto ágil (2001) se han desarrollado, mejorado, y adaptado metodologías de desarrollo de software que permiten a los equipos construir software de alta calidad con una inversión de esfuerzo bajo, poca documentación, lo que mejora la adopción de mejores prácticas de ingeniería de software (Abad,2012).

El desarrollo ágil de software es un conjunto de métodos y metodologías de desarrollo de software basado en el ciclo de vida de desarrollo de software iterativo e incremental, donde las necesidades y soluciones evolucionan a través de la colaboración entre la auto-organización y equipos multi-funcionales. Así mismo, promueve la planificación adaptativa, el desarrollo evolutivo y trabaja con la restricción del tiempo y la mayor productividad. Dentro de su política esta valorar:

- Individuos e interacciones sobre procesos y herramientas, software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

Las metodologías ágiles trabajan sobre iteraciones de plazos cortos fijos (timeboxes) que típicamente duran de una a cuatro semanas, que implican una planificación. Cada iteración incluye un equipo interdisciplinario de trabajo, funcional en todas las disciplinas de ingeniería de software: planificación, análisis de requisitos, diseño, codificación, pruebas unitarias y pruebas de aceptación, y despliegue. Al final de la iteración un producto de trabajo se presenta a las partes interesadas.

Esta forma de trabajo permite cambios rápidos, se obtiene una versión disponible (con un mínimo de errores) al final de cada iteración, la cual sea usable, testeable y valiosa para el cliente, en la que perciba un avance significativo.

6.4 Calidad del software relacionada con el producto

A través del tiempo la ingeniería de software ha tenido como propósito el desarrollo de metodologías que favorezcan la producción de software de calidad y esto se ha visto representado en normativas que estandarizan las buenas prácticas. Algunas de ellas son:

- **ISO/IEC 9126**

ISO 9126 la primera parte, denominada ISO 9126-1 es una extensión del trabajo previo realizado por McCall (1977), Boehm (1978), FURPS y otros en la definición de un conjunto de características de calidad de software. ISO9126-1 representa la última investigación sobre la caracterización de software para los fines de control de calidad, aseguramiento de la calidad y mejora de procesos de software (SPI).

ISO 9126 se desarrolló originalmente en 1991 para proporcionar un marco para la evaluación de la calidad del software y luego perfeccionado a lo largo de un período de más de diez años (Abran et al. 2003). Muchos estudios critican ISO 9126 por no prescribir los requisitos específicos de calidad, sino que la definición de un marco general para la evaluación de la calidad del software (Valenti 2002). Este es uno de sus puntos fuertes, ya que es más flexible y se puede utilizar en muchos sistemas, incluyendo los sistemas de e-learning. El modelo original define seis características del producto y estas a su vez se subdividen en una serie de sub-características. Ver cuadro 2.

Cuadro 2 Características y sub - características de la norma ISO 9126-1 (Fuente: Construcción propia)

Características	Sub características	Descripción
Funcionalidad	Idoneidad	Cumple la funcionalidad del software
	Exactitud	Relaciona con la precisión de la funcionalidad
	Interoperabilidad	Capacidad de interacción entre los componentes de un sistema.
	Conformidad	Se refiere a la capacidad de software compatible
	Seguridad	Controles para el ingreso al software
Confiabilidad	Madurez	Menor probabilidad de falla del software
	Tolerancia a fallos	Capacidad de soportar y recuperar
	Recuperabilidad	Traer de vuelta los datos y conexión a la red después de un fallo.
Usabilidad	Comprensibilidad	Facilidad de funcionamiento para el usuario
	Facilidad de aprendizaje	Esfuerzo de aprendizaje para los distintos usuarios de acuerdo a sus habilidades.
	Operatividad	La capacidad del software para ser manejado fácilmente por un usuario dado en un entorno determinado.
Eficiencia	Comportamiento Tiempo	Relacionado con los tiempos de respuesta del software
	Comportamiento de los recursos	Relacionado con el uso de recursos del sistema: memoria , disco duro entre otros.
Mantenibilidad	Analizabilidad	Capacidad de identificar la causa de un fallo
	Variabilidad	Cantidad de esfuerzo para cambiar un sistema
	Estabilidad	Sensibilidad al cambio de un sistema
	Capacidad de prueba	Esfuerzo necesario para realizar las pruebas a un sistema
Portabilidad	Adaptabilidad	Capacidad del sistema para cambiar a nuevas especificaciones o entornos
	Capacidad de instalación	Esfuerzo necesario para instalar el sistema
	Conformidad	Cumplimiento de la funcionalidad pero en cuanto a lo portable del sistema
	Reemplazabilidad	Facilidad para hacer intercambios de componentes en un entorno específico

- **ISO/IEC 14598 Software Product Evaluation**

Esta norma brinda las pautas para realizar el proceso de evaluación teniendo en consideración los posibles actores que se pueden tener como son los desarrolladores, los evaluadores, o compradores.

- **ISO/IEC 25000 Product Quality & Evaluation**

Esta familia de normas ISO/IEC 25000, proporciona una guía para el uso de la nueva serie de estándares internacionales llamada Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE - System and Software Quality Requirements and Evaluation). Constituye una serie de normas basadas en ISO/IEC 9126 y en ISO/IEC 14598 cuyo objetivo principal es guiar el desarrollo de los productos de software mediante la especificación de requisitos y evaluación de características de calidad.

Para controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición, ya que, como bien plantea Tom De Marco, "usted no puede controlar lo que no se puede medir". (Machinery, 1999)

Las cualidades para medir la calidad del software son definidas por innumerables autores, existen métricas, criterios y hasta nivel de complejidad del software y definen dos categorías de métricas: de complejidad de programa o código, y de complejidad de sistema o estructura.

La Calidad Sistémica de una organización es el reflejo de la calidad de sus productos de software y la calidad del proceso de desarrollo de los mismos (calidad de diseño y fabricación) de ahí que un buen proceso se represente en la calidad del producto.

6.5 Calidad relacionada con las personas

El éxito del producto software está dado por la calidad que den las personas a este proceso de desarrollo y estos sumados a la organización completa, de ahí que el centro de esta investigación se encuentre en la formación de los estudiantes desde sus comienzos, donde PSP se ajusta a esta necesidad. El reto está en cómo los analistas y programadores tienen buenas prácticas, de ahí que sea pertinente esta investigación porque se está trabajando con los estudiantes que apenas comienzan.

6.5.1 PSP/TSP

“Actualmente desarrollar un trabajo competente requiere la aplicación de buenas prácticas, planes y procedimientos que lleven de forma ordenada y eficiente las actividades y que permita a las personas concentrarse para construir productos de la más alta calidad. En este sentido, para desarrollar la industria Colombiana de Software es necesario contar con recurso humano altamente competitivo, entrenado en una disciplina personal y de trabajo en equipo” (Cardona, 2012).

Colombia se está preparando para ser potencia en la industria del Sector Software &TI a nivel mundial, a través de la capacitación oficial en PSP/TSP del Instituto de Ingeniería de Software (SEI) de la Universidad de Carnegie Mellon a 1047 personas a nivel nacional con el fin de mejorar sus niveles de productividad, lograr altos niveles de calidad y administrar eficientemente los proyectos en las empresas. (Fedesoft, 2012).

PSP traducido como Personal Software Process, fue propuesto por Watts Humphrey en 1995, es una metodología definida por cada persona, que controla, gestiona y mejora el trabajo en forma evolutiva. PSP permite hacer y cumplir compromisos personales, favorece el manejo de los tiempos generando autodisciplina y por ende cambios en el desarrollo del trabajo.

El desarrollo del (TSP) toma sus bases en la estrategia de calidad que propuso W. Edwards Deming (1982), con las etapas de planear, hacer, verificar y actuar. Y J.M. Juran (1988).

PSP concibe 3 fases de implementación:

- Planeación (plan de acuerdo a estimación)
- Desarrollo (definición de requisitos, diseño del programa, revisar el diseño y

corregir todos los defectos, codificar el programa , revisar el código y corregir todos los defectos, compilar y corregir todos los defectos, probar el programa y corregir todos los defectos).

- Posmorten (comparar los resultados reales con el plan, registrar los datos históricos del proceso, elaborar informe, documentar ideas para mejorar los procesos).

El asunto de estimación de tamaño y tiempo de un proyecto es una habilidad que se forma cuando hay disciplina en el proceso y se revisan los históricos, esto es posible cuando el talento humano trabaja de esta manera.

El SEI ha identificado que implementar PSP/TSP logra niveles de calidad 17 veces superiores a compañías con nivel 5 de madurez (CMMI), esfuerzo que se ve invertido en corrección de defectos 10 veces menor y variación en calendario 8 veces menor a los proyectos típicos de la industria.

En Colombia, una práctica que está generando resultados efectivos en las empresas es la de asignar “mentores” a los empleados nuevos, para orientarlos en su trabajo en un período que puede ser de varios meses o años; generalmente los mentores son personas de bastante experiencia en la empresa, ubicados en cargos de alto nivel jerárquico, con capacidad de comunicación, empatía, voluntad y orientación hacia las personas.

De acuerdo con The Coaches Training Institute (CTI) debe desarrollarse en los directivos y mandos medios, desde el nivel senior hasta el júnior, habilidades de dirección con la filosofía de “coaching”, es una de las prácticas acertadas que ha tomado impulso en los últimos tiempos en todo tipo de organizaciones, lo que genera liderazgo para el manejo de trabajo en equipos de alta exigencia y planes de desarrollo humano que buscan hacer competentes plenamente a todos sus empleados para el cargo actual.

Estas prácticas se acompañan de una enseñanza por problemas que intercala la práctica y la teoría y la capacitación por competencias donde se permite a la persona acoplar mejor sus atributos y las capacidades personales con las necesidades de formación a través de la evaluación del desempeño ayuda a identificar las brechas entre los comportamientos deseados y la realidad, lo que puede convertirse en los objetivos de la formación. (Fiti, 2013).

Al finalizar los niveles PSP viene Team Software Process (TSP) guía a los equipos de trabajo que están desarrollando productos de software, para hacerlo de forma segura, fiable, en menos tiempo y con menores costes.

El rendimiento del equipo se puede mejorar a través de un enfoque en las fortalezas y debilidades del equipo asociadas a las prácticas propias de TSP, de esta manera las prácticas del equipo se vuelven más fuertes y se logra que el rendimiento del equipo sea más sostenible.

6.5.2 SCRUM

Es un marco de trabajo de procesos del cual se pueden emplear varias técnicas que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Posee una filosofía frente a los equipos Scrum, roles, eventos, artefactos y reglas asociadas que permiten su éxito.

Se trabaja por Sprint, bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto "Terminado", utilizable y potencialmente desplegable. Identificando mejoras en su proceso de construcción, y en su forma de relacionarse como equipo, con el Scrum Master y el Product Owner. El equipo es auto organizado y multifuncional tiene todas las competencias necesarias para llevar a cabo el trabajo con flexibilidad, creatividad y productividad.

6.5.3 Metodología XP programación extrema

La programación extrema XP es considerada el método ágil más conocido y ampliamente utilizada. El nombre de XP fue acuñado por Beck (2000), debido a que el enfoque fue desarrollado utilizando las mejores prácticas del desarrollo iterativo y con la participación extrema del cliente.

En este método todos los requerimientos se expresan como escenarios (llamados historias de usuario), los cuales se implementan directamente como una serie de tareas.

“Los programadores trabajan en parejas y desarrollan pruebas para cada tarea, antes de escribir el código. Todas las pruebas se deben ejecutar satisfactoriamente cuando el código nuevo se integra al sistema” (Cendeja, 2014, p 101), por lo tanto se debe revisar constantemente el software, buscar posibles mejoras e implementarlas de forma inmediata.

7 PLANTEAMIENTO METODOLÓGICO

Para desarrollar la investigación se hace uso de la modalidad de investigación exploratoria descriptiva, consiste en elaborar una propuesta viable que atiende a necesidades en una organización que se han evidenciado a través de una investigación de campo (Tamayo, 2000). Se elige este tipo de investigación, con el fin realizar un proceso exploratorio para construir un conjunto de las mejores prácticas para el desarrollo de software empleadas por las empresas que hacen parte de Intersoftware, seleccionando las que pueden ser empleadas por los estudiantes de media técnica de las instituciones de educación media y diseñando un instrumento o estrategia que permita implementar estas mejores prácticas en el proceso formativo de los estudiantes.

La investigación utiliza como fuentes de información:

- Fuentes primarias: estudio de campo que permita recolectar de algunas empresas vinculadas a Intersoftware las mejores prácticas de desarrollo de software implementadas en sus organizaciones.
- Fuentes Secundarias: investigación documental de materiales ya elaborados acerca de las mejores prácticas para el desarrollo de software.

Con esta investigación se busca diseñar una estrategia/instrumento que permita implementar en el proceso de enseñanza aprendizaje de los estudiantes de media técnica en Informática, las mejores prácticas para el desarrollo de software permitiendo que los estudiantes vivan este proceso no solo en el desarrollo de software sino en sus procesos universitarios y en su diario vivir.

7.1 Fases del proyecto

La investigación se desarrolla en tres fases:

7.1.1 Fase 1. Contextualización de las mejores prácticas para el desarrollo de software

Esta fase tiene como objetivo identificar un conjunto de las mejores prácticas de desarrollo de software usadas por algunas de las empresas de Intersoftware.

Las actividades involucradas en esta etapa son:

- Identificación de actores y roles
- Diseño y aplicación de instrumentos de recolección de información
- Tabulación y análisis de la información recolectada

7.1.2 Fase 2. Establecer un conjunto de mejores prácticas de desarrollo de software para los estudiantes de media técnica

El objetivo de esta fase es seleccionar el conjunto de mejores prácticas para el desarrollo de software que los estudiantes de la media técnica deben implementar en su formación, cumpliendo las competencias que debe desarrollar.

Las actividades a desarrollar dentro de esta etapa son:

- Identificar dentro de la encuesta aplicada las prácticas de mayor repitencia y éstas cruzarlas con las correspondientes al perfil de los estudiantes.

7.1.3 Fase 3. Diseñar un instrumento que permita la implementación de las mejores prácticas de desarrollo de software en el proceso formativo de los estudiantes de media técnica

El objetivo es diseñar un instrumento o estrategia que permita la implementación de las mejores prácticas en el proceso formativo de los estudiantes de la media técnica de las que se seleccionaron.

Las actividades planteadas son:

- Plantear un instrumento que permita la implementación de las mejores prácticas de desarrollo de software en los estudiantes de media técnica.

7.2 Fuentes de información y análisis

Para realizar la investigación se toma como población las empresas que hacen parte de Intersoftware y del programa mentores empresariales y como muestra para aplicar los instrumentos y recolección de información, algunas empresas de dicha

agremiación, empresas dedicadas al testing, fábricas de software, especializadas en consultoría e inteligencia de negocios de TI, en seguridad informática y consultores expertos certificados en las principales plataformas tecnológicas del mercado: IBM, SAP, Oracle, J2EE, SUN, Microsoft y Open Source.

Intersoftware en la actualidad tiene 25 empresas asociadas y ésta hace parte de la Alianza Futuro Digital y por tanto se encuentra comprometida en el proceso de encadenamiento entre la educación media, la superior y el ámbito laboral para continuar fortaleciendo los lazos y trabajando sobre las debilidades del proceso encontradas en la media técnica.

Se tomó una muestra de 7 empresas que representan un 28 % (5 asociadas a Intersoftware y 2 no asociadas) estas conocen el trabajo que se viene desarrollando desde el año 2006 con la articulación de la educación media, técnica y tecnológica, y le apuestan a dicho proceso.

7.2.1 Técnicas de recolección de información

Para desarrollar la investigación se emplearon las siguientes técnicas de recolección de información:

- Técnica documental: se realiza una indagación y registro organizado de las mejores prácticas de desarrollo de software.
- Técnica de campo: se realiza entrevistas para obtener información acerca de las mejores prácticas de desarrollo de software empleadas en algunas de las empresas de desarrollo de software que pertenecen a Intersoftware.

Se realizan entrevista a empresarios del sector del software con preguntas de tipo abierta, cerrada y semicerrada, las cuales se realizan de forma presencial. Estas entrevistas pretenden:

- Identificar si las distintas certificaciones de calidad que obtienen las empresas aportan ventajas al desempeño del talento humano.

- Identificar las mejores prácticas implementadas en las empresas para que el personal desarrolle un buen producto/servicio software.
- Identificar cuál es el proceso de inmersión de los jóvenes que salen de la media técnica y/o técnica para incorporarse en la empresa.

7.2.2 Análisis de datos

Después de definir y aplicar las técnicas de recolección de datos, para realizar el análisis de estos se emplea el enfoque cuantitativo, a través del análisis de las respuestas obtenidas en cada una de ellas.

8 DESARROLLO METODOLÓGICO

8.1 Fase 1. Contextualización de las mejores prácticas para el desarrollo de software

8.1.1 Identificación de actores y roles

De acuerdo a la Alianza Futuro Digital Medellín se consideran:

- **Estudiantes de media técnica:** Joven entre los 14 y 17 años matriculado en una Institución de educación Media que optó por escoger una formación por ciclos propedéuticos.
- **Docente articulado:** Persona vinculada por la Secretaria de Educación que acompaña y apoyan a los docentes articuladores de las IES en la operación de los módulos de los programas transformados y que corresponden a los saberes de la disciplina del desarrollo de software.

- **Docente articulador:** Persona vinculada por la Institución de Educación Superior que acude a la Institución de educación media (IEM) para coordinar, implantar y supervisar conjuntamente con el docente de ésta, la ejecución de las distintas secuencias programáticas del correspondiente módulo, y el aprendizaje integral de los estudiantes.
- **Asesor de PPI:** El asesor de los Proyectos Pedagógicos Integradores (PPI) desarrollados por los estudiantes de media técnica en cada una de las Instituciones Educativas, tiene las siguientes funciones:

Desempeñar el rol de “Cliente” en el Proyecto Pedagógico Integrador, de forma que los alumnos se familiaricen en la interacción de proveedor de servicios con su “Cliente”. Asesorar y supervisar el cumplimiento de la planeación y la coordinación del plan de trabajo de cada equipo de alumnos de cada IEM que están desarrollando su respectivo PPI.

- **Empresas de Intersoftware:** Intersoftware es una corporación sin ánimo de lucro, creada en abril de 2004 por iniciativa del sector productivo, a la fecha está conformada por 29 empresas de software, las cuales están clasificadas en 10 categorías, las cuales son:
 - Aseguramiento de la calidad y testing
 - Desarrollo a la medida
 - Capacitación y entrenamiento
 - Agencia digital
 - Desarrollo de software empaquetado
 - Servicios de pruebas de software
 - Servicios de soporte IT - Soluciones BI
 - Outsourcing
 - Consultoría TI

- Aseguramiento de la información

El perfil de formación de los jóvenes de la media técnica (programación) aunque se vincula con todas las categorías de empresas antes mencionadas, tiene relación directa con Desarrollo de software a la medida y empaquetado. De las categorías Outsourcing, Consultoría TI y Aseguramiento de la información no hay representación de empresas lo cual no resulta tan significativo respecto a su campo de acción para este proyecto.

La experiencia de las empresas entrevistadas está representada en un promedio de 18 años en el mercado lo que representa una trayectoria importante y autoridad en el tema del desarrollo de software. Ver cuadro 3.

Cuadro 3 Características empresas analizadas (Fuente: Construcción propia)

Categoría	Empresa	Año Fundación
Capacitación y entrenamiento Desarrollo de software empaquetado	Ilimitada SAS	1.986
Servicios de pruebas de software	Choucair Testing SA	1.999
Agencia digital Desarrollo a la medida	Pragma S.A	1.996
Servicios de soporte IT Soluciones BI	MVM	1.995

Categoría	Empresa	Año Fundación
Desarrollo a la medida: Soluciones inteligentes para el sector de tránsito y transporte	Quipux	1.995
Desarrollo a la medida: Servicios de información de mercados.	Mercadeo Virtual	1.995
Aseguramiento de la calidad y testing Capacitación y entrenamiento	V&V Quality	2.006

8.1.2 Diseño y aplicación de instrumentos de recolección de información

La identificación de las prácticas sobre las cuales se preguntaría a los empresarios fue el fruto del cruce de los componentes relacionados con la calidad en el proceso y en el producto. Ver cuadro 4.

Cuadro 4 Relación proceso – producto software (Fuente: Construcción propia)

Proceso de software	Producto de software					
	Funcionalidad	Fiabilidad	Usabilidad	Eficiencia	Mantenibilidad	Portabilidad
Proceso de adquisición de sistema o producto de software	X					
Determinación de Requerimientos	X	X	X	X		X
Proceso de Validación	X	X		X		X
Proceso de Revisión Conjunta	X		X			
Proceso de Operación		X				

Proceso de software	Producto de software					
	Funcionalidad	Fiabilidad	Usabilidad	Eficiencia	Mantenibilidad	Portabilidad
Proceso de Verificación		X				
Proceso de Documentación			X		X	
Proceso de Desarrollo					X	X
Gestión de configuración					X	
Proceso de Gestión de la calidad	X	X	X	X	X	X
Proceso de Aseguramiento de la calidad	X	X	X	X	X	X
Proceso de auditoria	X	X	X	X	X	X

Luego las prácticas clasificadas en organizacionales y las relacionadas con los procesos de desarrollo: requisitos, análisis – diseño y desarrollo fueron verificadas a la luz de las normas de calidad que tienen que ver con el proceso ver columnas

Fuente a partir de la cuadro 5a.

Cuadro 5a Relación prácticas organizacionales y su fuente (Fuente: Construcción propia)

Etapas de Desarrollo Software	N°	Descriptor práctica Organizacional	Fuente							
			ISO 12207:2008 Procesos ciclo de vida	ISO 15504 Evaluación nivel de madurez	Proceso Unificado de Desarrollo de Software	CMMI for Development	TSP	PSP	PMBOK - IEEE Std 1490-2003	Met. ágiles SCRUM
Todas	1	El equipo debe conocer y aceptar los objetivos del proyecto	X	X	X	X	X	X	X	X
Todas	2	El equipo esta auto organizado					X			X
Todas	3	El equipo debe conocer y aceptar las tareas necesarias para completar cada requisito	X	X	X	X	X	X	X	X
Todas	4	Estimar el esfuerzo necesario para realizar cada tarea.	X	X	X	X	X	X	X	X
Todas	5	Cada miembro del equipo se auto asigna a las tareas.					X			X
Todas	6	Cada miembro del equipo tiene orientación al logro.	X	X	X	X	X	X	X	X
Todas	7	Conocer las habilidades de cada miembro del equipo para identificar su aporte					X	X	X	X
Todas	8	Dedicación de tiempo en sus tareas por parte de cada miembro	X	X	X	X	X	X	X	X
Todas	9	Reuniones diarias de seguimiento de equipo no mayores a 15 minutos y preguntarse ¿Que hice ayer, Que haré hoy y que dificultades tengo					X	X		X
Todas	10	Destinación de reuniones de retroalimentación con el líder proyecto(docente)	X	X	X	X	X	X	X	X
Todas	11	Revisar el trabajo de otros en forma objetiva, sincera y propiamente documentada.	X	X	X	X	X	X	X	X
Todas	12	Escuchar equitativamente las opiniones, preocupaciones y quejas de un colega.	X	X	X	X	X	X	X	X
Todas	13	En situaciones fuera de sus propias áreas de competencia, solicitar las opiniones de otros profesionales que tengan competencia en esa área.	X	X	X	X	X	X	X	X
Todas	14	Mejorar su habilidad para producir documentación precisa, informativa y bien redactada.	X	X	X	X	X	X	X	X
Todas	15	Se realiza control de cambios	X	X	X	X	X	X	X	X
Todas	16	Se realiza modelado visual del software, documento escrito y auditivo que permita la comprensión de la información	X	X	X	X	X	X	X	X
Todas	17	Se realiza revisión entre pares	X	X	X	X	X	X	X	X
Todas	18	Documentar lecciones aprendidas	X	X	X	X	X	X	X	X

Cuadro 7 Relación prácticas análisis – diseño y su fuente (Fuente: Construcción propia)

Norma de competencia	Elementos de competencia	N°	Descriptor práctica de Análisis - Diseño	Fuente							
				ISO 12207:2008 Procesos ciclo de vida	ISO 15504 Evaluación nivel de madurez	Proceso Unificado de Desarrollo de Software	CMMI for Development	TSP	PSP	PMBOK - IEEE Std 1490 2003	Met. ágiles SCRUM
Analizar los requerimientos del cliente para construir la solución	Bosquejar el modelo funcional de acuerdo con la información recolectada y la metodología seleccionada	1	Comprender bien el concepto de arquitectura por capas.	X	X	X	X	X	X	X	X
		2	Alto nivel de abstracción para favorecer discusiones sobre diversos niveles y soluciones arquitectónicas para la solución		X	X	X	X	X	X	X
		3	El diseño general del sistema se construye conforme avanza el desarrollo del proyecto cambiando y evolucionando constantemente	X	X	X	X	X	X	X	X
	Detallar los requerimientos expresados en el modelo funcional de acuerdo con la metodología seleccionada	4	Los modelos de diseño no se encuentran completos, los detalles se refinan durante el proceso de codificación.	X	X	X	X	X	X	X	X
		5	Los estudiantes deben dibujar las interfaces siguiendo una lista de chequeo		X	X	X	X	X		X
		6	Identificar los requisitos de interfaz.		X	X	X	X	X	X	X
		7	Analizar y validar los requisitos.	X	X	X	X	X	X	X	X
	Detallar la estructura técnica de acuerdo con el análisis de los	8	Establecer los conceptos y los escenarios de operación		X	X	X	X	X	X	X

Norma de competencia	Elementos de competencia	N°	Descriptor práctica de Análisis - Diseño	Fuente							
				ISO 12207:2008 Procesos ciclo de vida	ISO 15504 Evaluación nivel de madurez	Proceso Unificado de Desarrollo de Software	CMMI for Development	TSP	PSP	PMBOK - IEEE Std 1490 2003	Met. ágiles SCRUM
Documentar procesos y recursos de los sistemas de información	requisitos de la solución a construir	9	Establecer una definición de la funcionalidad y de los atributos de calidad requeridos.	X	X	X	X	X	X	X	X
	Describir la evolución de los procesos de la aplicación de acuerdo con los estándares y las normas establecidas	10	Analizar detenidamente el ambiente de implementación.	X	X	X	X	X	X	X	X
	Construir los manuales para sistema de información de acuerdo con las normas establecidas	11	Los diseñadores también deben codificar.		X		X	X	X	X	X
12		Cuando el modelo es desarrollado por alguien más se entrega a otra persona para codificarlo hay un riesgo significativo de que no capte sus detalles adecuadamente. Separar las funciones del diseño y la codificación es riesgosa, es mejor contar con especialistas en el equipo que puedan diseñar y codificar.		X		X	X	X	X	X	

Cuadro 8b Relación prácticas desarrollo y su fuente (Fuente: Construcción propia)

Norma de competencia	Elementos de competencia	N°	Descriptor práctica de Desarrollo	Fuente							
				ISO 12207:2008 Procesos ciclo de vida	ISO 15504 Evaluación nivel de madurez	Proceso Unificado de Desarrollo de Software	CMMI for Development	TSP	PSP	PMBOK - IEEE Std 1490-2003	Met. ágiles SCRUM
		6	Documentar las partes complicadas del sistema. A través de la documentación generada debemos poder entender el funcionamiento del sistema, así como las razones que sustentan las decisiones de diseño.	X	X	X	X	X	X	X	X
	Construir los manuales para sistema de información de acuerdo con las normas establecidas	7	Verificación y seguimiento de estándares de codificación.					X	X		X
		8	Adecuada planificación de la construcción y mantener mediciones acerca del desempeño y calidad de los programadores	X	X	X	X	X	X	X	X
		9	Usar con propiedad los diferentes lenguajes que apoyan el proceso de construcción en los diferentes niveles de la arquitectura: lenguajes de presentación, lenguajes de programación, lenguajes de configuración, lenguajes de bases de datos					X	X		X
Norma de competencia	Elementos de competencia	N°	Descriptor práctica de Desarrollo	ISO 12207:2008 Procesos ciclo de vida	ISO 15504 Evaluación nivel de madurez	Proceso Unificado de Desarrollo de Software	CMMI for Development	TSP	PSP	PMBOK - IEEE Std 1490-2003	Met. ágiles SCRUM
Documentar procesos y recursos de los sistemas de información	Documentar la administración del software de acuerdo con normas y estándares establecidos.	10	Contabilizar el tiempo que se invierte para desarrollar determinada funcionalidad permitirá tener claridad de ese tiempo para desarrollar otra funcionalidad.						X		
		11	Los defectos deben registrarse tan pronto se identifican		X		X	X	X	X	X
		12	Programar bien con buenas prácticas y con expectativas de aprender		X			X	X		X

Para recoger la información se diseñó una entrevista semi estructurada, con una encuesta que contiene preguntas cerradas, semi abiertas y abiertas (ver anexo 1) con el fin de tener un encuentro personal con el gerente de la empresa o en su defecto el gerente de proyectos, que permitió tomar información confiable y otra adicional que aportó a la investigación de acuerdo con las relaciones planteadas en las Cuadros 5 a 8b.

La encuesta para los empresarios está compuesta por 3 componentes: Organizacional, Talento humano y Buenas prácticas para desarrollo del software. Ver cuadro 9.

*Cuadro 9 Variables a evaluar en las empresas a través de la encuesta
(Construcción propia)*

COMPONENTE	VARIABLES A EVALUAR
Organizacional	
Pretende conocer los tipos de certificación que poseen las empresas y reconocer a partir de ello las ventajas para el desarrollo del producto/servicio software que brindan. Además identificar las buenas prácticas organizacionales que usan y que han permitido su permanencia en el mercado.	Certificación vs desarrollo del producto/servicio software.
Talento humano	
Indaga acerca del proceso de inducción al personal nuevo que ingresa a la empresa, incluyendo personas formadas en la media técnica, identificando las labores que realizan y el grado de desarrollo de competencia con que llegan al mundo laboral.	Nivel de desarrollo de las competencias con que llegan los estudiantes egresados de la media técnica a la empresa.
Proceso desarrollo de Software	
Identificar las buenas prácticas que las empresas implementan categorizadas en Organizacionales - Requisitos - Análisis y diseño - Desarrollo.	Buenas prácticas que más se repiten en los distintos procesos en las empresas.

A continuación se describen los principales hallazgos encontrados después de realizar las diferentes entrevistas, teniendo en cuenta cada uno de los componentes con los cuales se estructuró la entrevista: Organización, talento humano y proceso de desarrollo de software.

8.1.3 Componente Organizacional

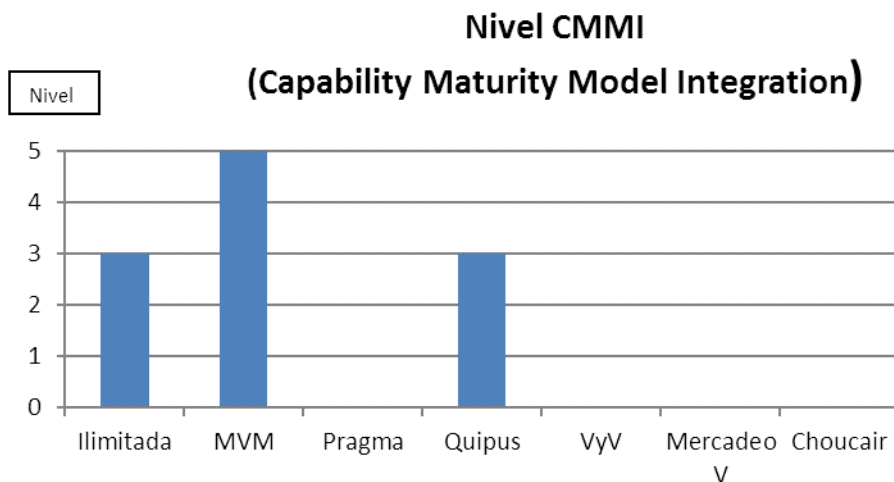
A partir de la recolección de la información a través de las entrevistas se encuentra que las empresas analizadas tienen las siguientes certificaciones:

*Cuadro 10 Certificaciones obtenidas por las empresas analizadas – sector software
(Construcción propia)*

EMPRESA	CERTIFICACIONES
Ilimitada SAS	CMMI ISO 9001
Choucair Testing SA	ISO 9001 ISTQB
Pragma S.A	ISO 9001 TSP PSP SCRUMM
MVM	CMMI ISO 9001
Quipux	CMMI ISO 9001 27000 CMMI servicios
Mercadeo Virtual	ninguna
V&V Quality	ISO 9001

De las 7 empresas entrevistadas, 3 (2 con nivel 3 y 1 con nivel 5) poseen certificación CMMI Modelo de Madurez de Capacidades (Capability Maturity Model Integration), lo que representa un 42.8 % de las empresas que toman la opción de controlar los procesos y los productos a través de este marco de mejora, son empresas más dedicadas al desarrollo de software. (Ver Ilustración 3).

Ilustración 3 Nivel CMMI implementado por las empresas analizadas (Construcción Propia)



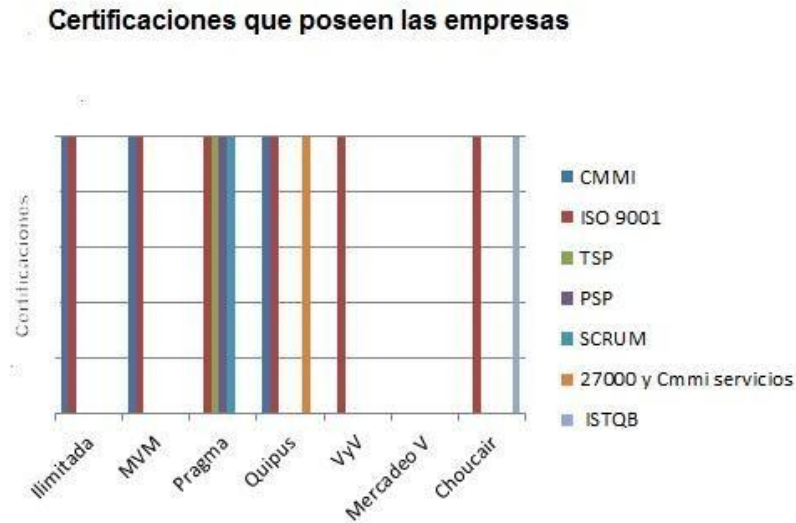
Para ISO 9001 el 85,71 % poseen certificación en Sistemas de calidad por más de 5 años, teniendo en cuenta que es un proceso general que viene desde 1.987 en su versión original y primera revisión en 1.994.

En general se observa que la certificación en Sistemas de calidad es la más frecuente y fortalecida en las empresas, generando organización en el ambiente de trabajo y haciendo una revisión de los procesos generales de la empresa.

Pragma es la única de las empresas entrevistadas que posee la certificación en TSP (Team Software Process) – PSP (Personal Software Process) y Scrum (Marco para desarrollo ágil).

Quipus es la única con ISO/IEC seguridad de la información 27000, CMMI servicios y Choucair con ISTQB (Certifying Software Testers Worldwide) como soporte de su modelo de negocio. Dichas certificaciones son logros obtenidos de acuerdo a las características del negocio, sus intereses y necesidades, el tener una o más certificaciones hace parte de las políticas de la empresa y las exigencias del medio.

Ilustración 4 Certificaciones de calidad obtenida por las empresas analizadas (Construcción propia)



La empresa Mercadeo Virtual que no tiene certificaciones no considera aún la necesidad de hacerlo, sin embargo tiene políticas que favorecen el desarrollo de un servicio/producto de calidad. (Ver Ilustración 4).

De acuerdo a lo manifestado por los empresarios y a la evaluación de la variable: Certificación vs desarrollo del producto/servicio software, se concluye:

- Las certificaciones implican estandarización de los procesos y las empresas reconocen la importancia de ésta, para favorecer el trabajo de los equipos que se ve representado en productividad y calidad. Lo que implica el ingreso de personas en diferentes momentos y hacer ajustes fácilmente a los procesos.
- Se abren posibilidades para la empresa para participar en licitaciones y se gana mayor credibilidad y confianza en el proceso de calidad.
- El desarrollo del producto/servicio software tiene un proceso documentado que lo organiza y permite identificar recursos y oportunidades que posibilitan la mejora continua.
- Mejora los tiempos de respuesta y brinda calidad de vida para los empleados y por supuesto al cliente.
- Por otra parte los empresarios afirman que estar certificados no garantiza que todo esté bien, depende mucho de la voluntad del talento humano.

8.1.3.1 Componente Buenas Prácticas organizacionales

Al revisar las prácticas organizacionales que las empresas usan se encuentra que: Las prácticas de la # 1 a la # 5 se realizan A VECES.

1. Revisar el trabajo de otros en forma objetiva, sincera y propiamente documentada
2. Escuchar equitativamente las opiniones, preocupaciones y quejas de un colega.
3. Ayudar a sus colegas a que estén totalmente alertas a los actuales estándares incluyendo políticas y procedimientos de protección de contraseñas, archivos, información confidencial y las medidas de seguridad en general.
4. En situaciones fuera de sus propias áreas de competencia, solicitar las opiniones de otros profesionales que tengan competencia en esa área.
5. Mejorar su conocimiento de los avances en el análisis, especificación, diseño, desarrollo, mantenimiento, pruebas del software y documentos relacionados, junto con la administración del proceso de desarrollo

Las prácticas de la # 6 hasta la #16 son SIEMPRE usadas por todas las empresas.

6. Mejorar su habilidad para producir documentación precisa, informativa y bien redactada.
7. Mejorar su comprensión del software de los documentos con que se trabaja y del medio ambiente donde serán usados
8. Calidad en el proceso dará calidad en el producto
9. Asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
10. Uso de arquitectura basada en componentes
11. Control de cambios

12. Modelado visual del software
13. El aseguramiento de la calidad es parte del proceso de desarrollo y no la responsabilidad de un grupo independiente
14. Hay cultura organizacional, equipos auto-organizados, auto – control, auto – capacitación.
15. Documentar lecciones aprendidas
16. Se realiza revisión entre pares.

Para las prácticas #1 y # 3 el 14% No la realizan y el 86% la hacen siempre. Para las prácticas # 2, 4 y 5 14% No la realizan, 14% A veces la realizan y el 72% la hacen siempre.

Las anteriores prácticas que tienen variaciones están directamente relacionadas con el talento humano, comprometidas con el trabajo en equipo y los procesos de comunicación en la organización, esto se explica teniendo en cuenta la jerarquización de los roles que cumplen las personas dentro de las empresas, que hace que ciertas personas con determinado rol tomen las decisiones frente a otros. Cuando se solicitan tantas opiniones a las personas puede generarse incertidumbre y duda a la hora de la toma de la decisión.

Respecto a la práctica # 5: “Mejorar su conocimiento de los avances en el análisis, especificación, diseño, desarrollo, mantenimiento, pruebas del software y documentos relacionados, junto con la administración del proceso de desarrollo”.

Dicha práctica hace que se brinde tiempo a las personas que se encuentran comprometidas con estas etapas para acceder e interpretar la información, lo que podría representar tiempo considerable mientras que al especializar personas por etapas se favorecería en cuanto al tiempo, pero toda la responsabilidad estaría cargada a cada líder y se cerraría el potencial de aprendizaje y crecimiento de los otros equipos de trabajo.

8.1.4 Componente Talento Humano

8.1.4.1 Proceso de inducción al personal nuevo que ingresa a la empresa

Las empresas realizan proceso formal de inducción que dura entre 1 a 3 semanas en una (1) empresa está relacionada con Gestión del conocimiento y en las demás directamente con su labor en programación. Se realiza inducción inicial, luego acompañamiento y hasta la realización de examen práctico. No hay entrega de manual de funciones.

El promedio de tiempo que duran los jóvenes en la organización es de año y medio en la mayoría de los casos el retiro se debe a una mejor oferta de trabajo.

Un 72% de las empresas han contratado en su organización personas que estén realizando su formación por ciclos propedéuticos acabados de salir del colegio (Media técnica) y de igual manera en la actualidad cuenta con este tipo de personal.

8.1.4.2 Grado de desarrollo de competencia con que llegan los estudiantes al mundo laboral

En cuanto al grado de desarrollo de las competencias que se evidencia en el trabajo de los jóvenes, cada empresa valoró de 1 a 5 el desarrollo de las competencias y se calculó luego una media.

Frente al Trabajo en equipo, orientación al logro y manejo de TIC, se identifica un buen avance con una media de 3.5. Solo una de las empresas consultadas cuestionó estas competencias, y propone replantear el concepto de lo que es ser un buen programador.

Ilustración 5 Nivel de desarrollo: Trabajo en equipo (Construcción propia)



Ilustración 6 Nivel de desarrollo: Orientación al logro (Construcción propia)

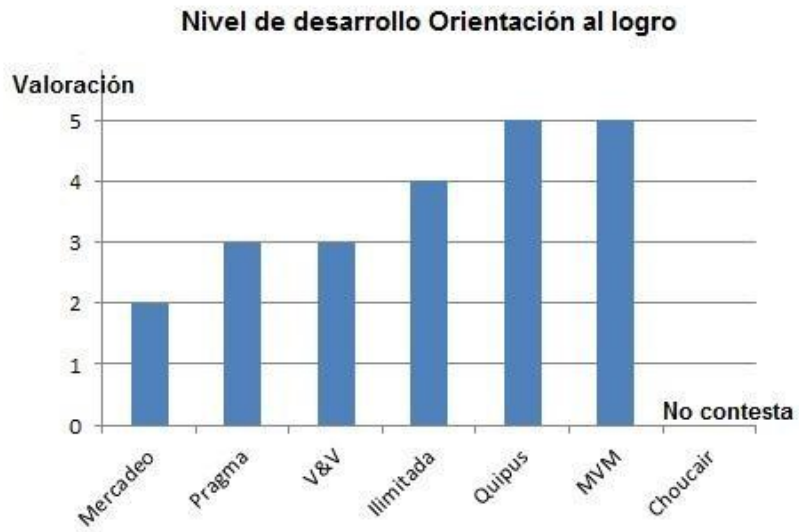


Ilustración 7 Nivel de desarrollo: Manejo de TIC (Construcción propia)



Luego aparecen en orden decreciente las competencias: *Programar en un lenguaje* con una media de 3.4, *Investigación y creatividad* con una media 3.3, *Comunicación oral y escrita* con una media 3.2, *Pensamiento Algorítmico* con una media 3.

Ilustración 8 Nivel de desarrollo: Programar en un lenguaje (Construcción propia)

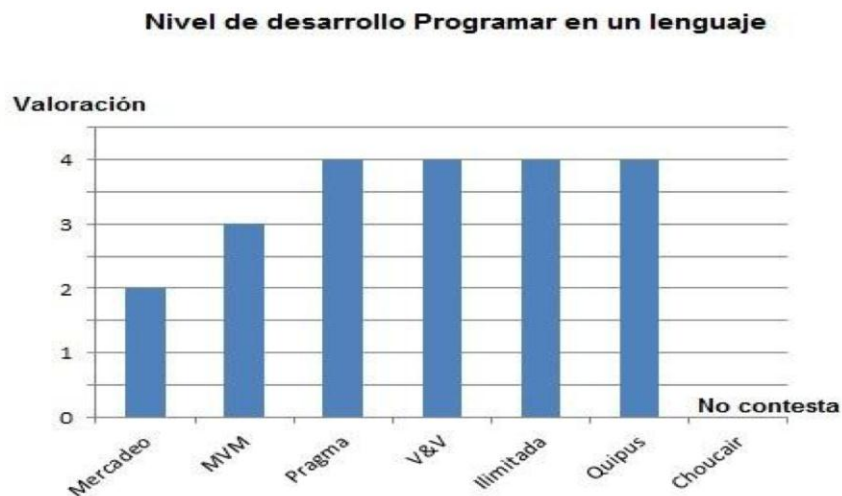


Ilustración 9 Nivel de desarrollo: Investigación y creatividad (Construcción propia)

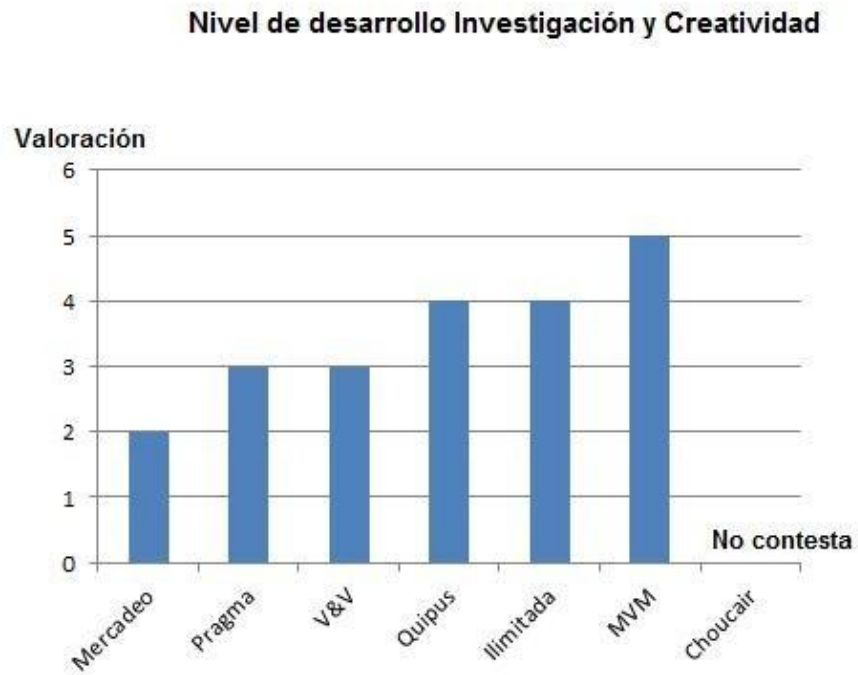


Ilustración 10 Nivel de desarrollo: Comunicación Oral y escrita (Construcción propia)

Nivel de desarrollo Comunicación oral y escrita

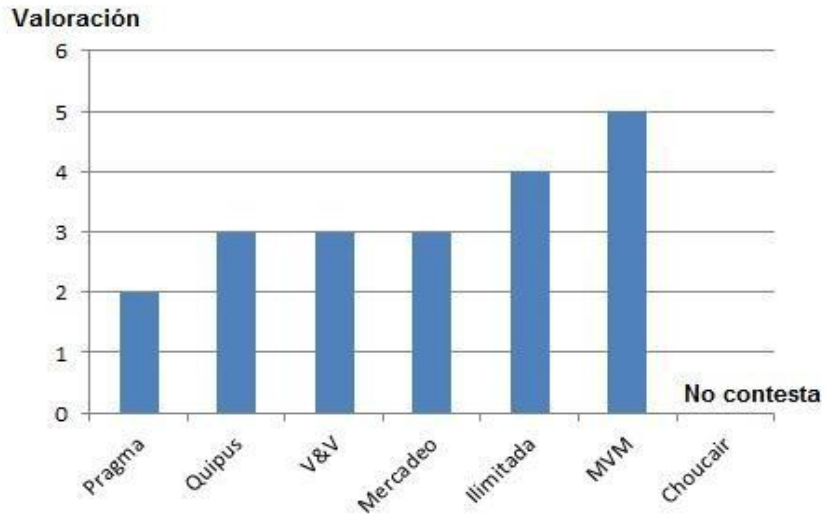
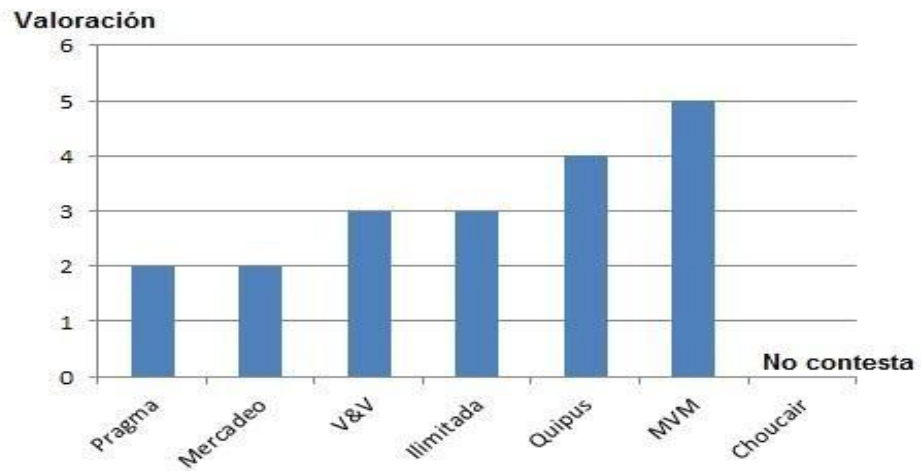


Ilustración11 Nivel de desarrollo: Pensamiento Algorítmico (Construcción propia)

Nivel de desarrollo Pensamiento Algorítmico



Los más bajos niveles de desarrollo están en Pensamiento analítico – sistémico con una media de 2.9, Documentación de un software con una media 2.6, Comprensión organizacional con una media 2.5 y por último Construcción de modelado de datos con una media de 2.1.

Ilustración 12 Nivel de desarrollo: Pensamiento analítico-sistémico (Construcción propia)

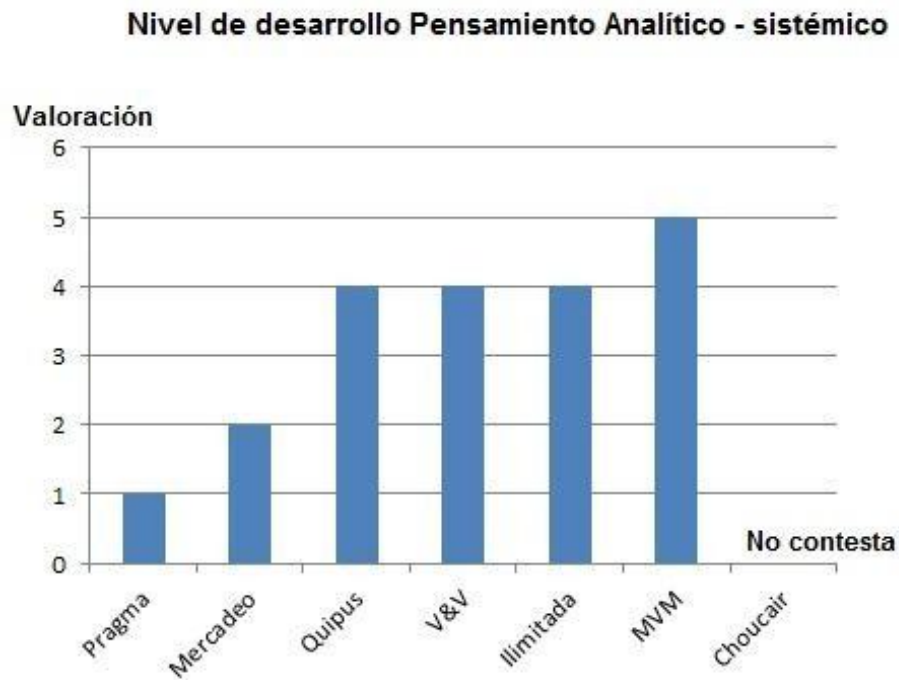


Ilustración13 Nivel de desarrollo: Documentación de un software (Construcción propia)



Ilustración 14 Nivel de desarrollo: Comprensión Organizacional (Construcción propia)

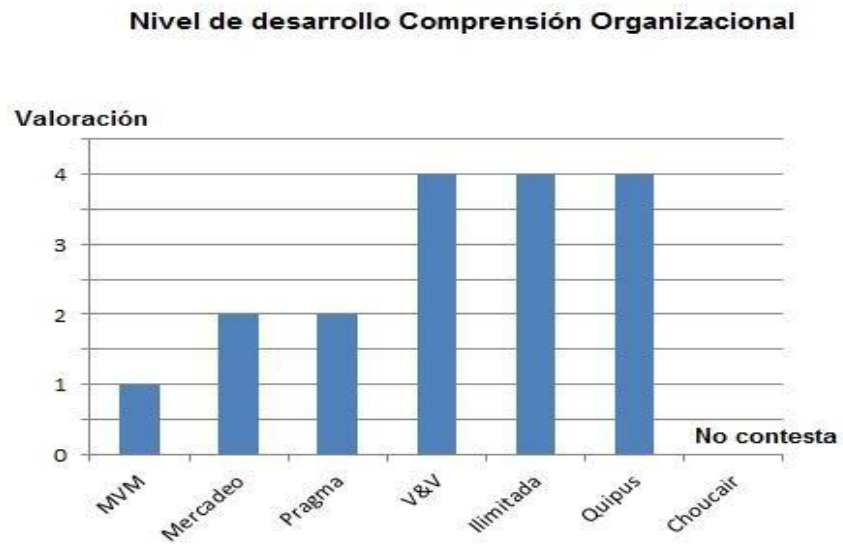


Ilustración15 Nivel de desarrollo: Construcción de modelado de datos
(Construcción propia)



En conclusión frente a la variable Nivel de desarrollo de las competencias con que llegan los estudiantes egresados de la media técnica a la empresa se puede decir que:

- El nivel más alto alcanzado lo tienen las competencias de Trabajo en equipo, Orientación al logro y manejo de TIC, las que se encuentran en un nivel medio de desarrollo son Programar en un lenguaje, Investigación y creatividad, Comunicación oral y escrita y Pensamiento Algorítmico.
- Las competencias con niveles más bajos son: Pensamiento analítico – sistémico, Documentación de un software, Comprensión organizacional y por último Construcción de modelado de datos.
- La lectura de esta información permite deducir que las competencias que se encuentran mejor hay que continuar trabajándolas, prestar mayor atención al desarrollo de las que se encuentran en nivel medio teniendo en cuenta que son competencias claves del perfil de salida de los estudiantes como lo es Programar en un lenguaje y el Pensamiento Algorítmico, de otro lado son

competencias básicas la Investigación, creatividad, Comunicación oral y escrita.

- Respecto a las competencias Pensamiento analítico – sistémico y documentación de un software hacen parte clave del perfil de salida de los estudiantes, y por lo tanto debe pensarse la manera de afianzarlas, en cuanto a la Comprensión organizacional el desarrollo del Proyecto Pedagógico Integrador (PPI) hace parte fundamental para comprender los contextos y por último Construcción de modelado de datos es una competencia que los empresarios consideran compleja de lograr, no muy acorde al perfil de Programación pero que cuenta con todo el sentido si se quiere lograr comprensión del conocimiento, más bien el reto está en la pedagogía para favorecer el aprendizaje.

8.1.4.3 Roles en las distintas etapas del proceso de desarrollo de SW

En las 7 organizaciones entrevistadas las tareas de toma de requisitos, análisis y diseño le son asignadas preferiblemente a los ingenieros (profesionales) con un 71.4% pero hay un 28.5%, cuya experiencia ha mostrado que estas tareas también puede participar activamente un técnico y/o un tecnólogo. (Ver Ilustración 16).

Respecto a las tareas de desarrollo que representan la principal salida ocupacional de los jóvenes de la media técnica, se tiene que el 44.4% de las empresas consideran que todos los perfiles están en la capacidad de generar codificar. El 22.2% de las empresas da esta tarea sólo a los tecnólogos, el 11.1% sólo a los técnicos, 11.1% sólo a los profesionales y el 11.1% no contesta. (Ver Ilustración 17).

Ilustración 16 Perfil profesional de desempeño en las fases de requisitos, análisis y diseño (Construcción propia)

Perfil profesional que se desempeña en Requisitos, Análisis y Diseño

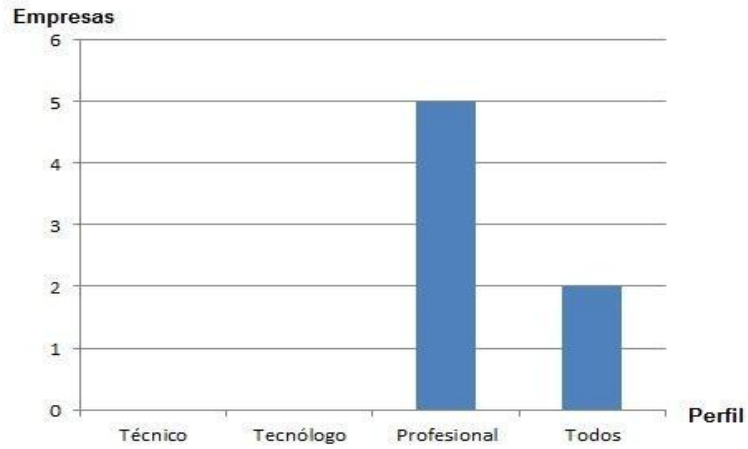
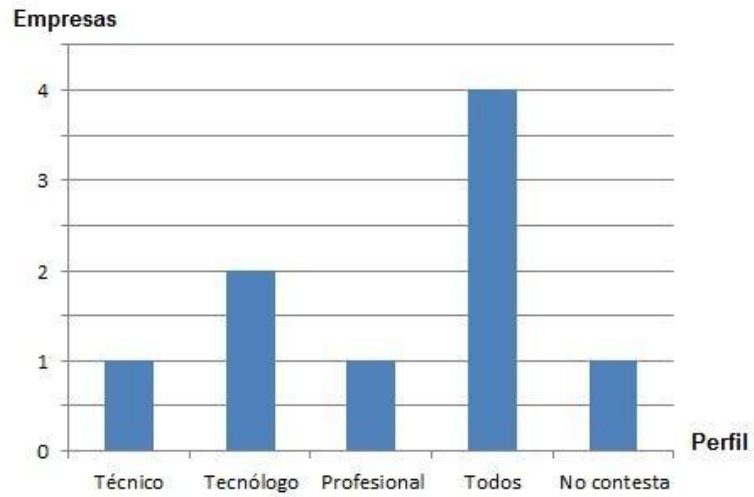


Ilustración 17 Perfil profesional de desempeño en la fase de desarrollo de software (Construcción propia)

Perfil profesional que se desempeña en Fase de Desarrollo de Software



En cuanto a las prácticas organizacionales el 44.4% de las empresas consideran que todos los perfiles deben participar activamente de esta cultura organizacional, el 33.3% consideran que sólo los profesionales y el 22.2% sólo los tecnólogos. (Ver Ilustración 18).

Ilustración 18 Perfil profesional de desempeño en prácticas organizacionales (Construcción propia)



Los empresarios entrevistados destacan dentro de las mejores prácticas implementadas para que el personal desarrolle un buen producto software, las siguientes:

- Realizar seguimiento y control al proceso
- Realizar medición del proceso.
- Interacción con el cliente en entregas tempranas
- Revisión de pares en documentos de identificación de errores
- Involucrar el equipo testigo desde el principio del proceso
- Cultura de mejora de procesos

- Usar procesos de automatización
- Trabajo de matemáticas discretas
- Incentivar la realización de Pruebas unitarias(probarse así mismo)
- La calidad del proceso está en la Ingeniería de Requisitos
- Programar bien con buenas prácticas y con expectativas de aprender
- Comprender bien el concepto de arquitectura por capas.
- Se hace reflexión continua frente a lo bueno, lo repetible y no repetible.
- Para facilitar un mejor ensamblaje de los procesos, lograr procesos comunicativos que afiancen la coherencia entre las personas.
- Todo se planea
- Incentivar el Control de versiones

Las empresas consideran que dentro de los cambios que deben hacerse para vincular a los jóvenes a la vida laboral son:

- Brindar información a los estudiantes frente a lo que implica la contratación en una empresa y el contexto organizacional.
- Realizar pruebas técnicas a los estudiantes donde se evalúen los avances.
- Tener una base de datos de los egresados de acuerdo a su potencial (cualitativa, cuantitativa, técnico)
- Ponerle motivación y pasión a lo que se hace.
- La buena inducción para identificar la vocación.
- Tener pasantías en las empresas que permitan conocer el mundo de la organización, invitarlos desde grado 9°
- El manejo técnico de los lenguajes de programación.

Debe continuarse trabajando con los jóvenes en:

- La buena comunicación oral y escrita
- El énfasis en el estudio de la segunda lengua
- Aclarar expectativas frente a lo que se van a enfrentar en el mundo de la empresa.
- El desarrollo del Pensamiento analítico – sistémico.

8.1.5 Componente Proceso de desarrollo de Software

Para este componente se analizan las buenas prácticas para las etapas de: Requisitos, análisis y diseño, desarrollo.

8.1.5.1 Etapa Requisitos

Se identificaron 7 prácticas:

1. Identificar, definir y atender asuntos éticos, económicos, culturales, legales y ambientales relacionados a los proyectos de trabajo.
2. Participación activa de los clientes.
3. Visualización de los requerimientos organizados por prioridad.
4. Comunicación asertiva con su equipo de trabajo, entrega de información de una sola fuente.
5. Visualización de la arquitectura, primero desde un nivel de abstracción alto para identificar una estrategia que permita la implementación de la solución.
6. Elaborar documentación entregable a través del ciclo de vida del producto de forma paralela a la creación de la solución.

7. Ver más allá del modelado. Algunas veces los requerimientos que tienen una mayor prioridad son complejos, lo cual implica explorarlos antes de comenzar su desarrollo para reducir el riesgo general del desarrollo

Se puede determinar que las prácticas 2 a 5 se practican SIEMPRE en las organizaciones entrevistadas, frente a la 1: "Identificar, definir y atender asuntos éticos, económicos, culturales, legales y ambientales relacionados a los proyectos de trabajo". El 66.6 % de las empresas consultadas afirman SIEMPRE practicarlas, existe un 16.6 % que la realiza AVECES y otro 16.6 % que NO lo hace, lo que permite pensar que no hay una cultura arraigada de análisis de riesgos que permita contemplar el proyecto en su integralidad.

En cuanto a la práctica 6: "Elaborar documentación entregable a través del ciclo de vida del producto de forma paralela a la creación de la solución" tiene un 83% en SIEMPRE y un 17% en A VECES representada en una empresa nivel 5 en CMMI que sería necesario revisar la razón esta práctica.

La práctica 7: "Ver más allá del modelado". Algunas veces los requerimientos que tienen una mayor prioridad son complejos, lo cual implica explorarlos antes de comenzar su desarrollo para reducir el riesgo general del desarrollo" tiene divididas las opiniones en un 50% en SIEMPRE y el otro 50% en A VECES lo que hace pensar en una práctica que atiende nuevamente a la prevención de los riesgos.

A esta categoría de prácticas relacionadas con los requisitos se agregan dos planteadas por los entrevistados: "Verificación que no se pasó algo por alto y Visita al cliente, en vivo y directo" son oportunas en la medida de tomar de la fuente (cliente) la información y revisar antes de seguir adelante con otra etapa para no pasarla por alto.

8.1.5.2 Etapa Análisis y diseño

Se identificaron 5 prácticas:

1. El diseño general del sistema se construye conforme avanza el desarrollo del proyecto cambiando y evolucionando constantemente
2. Los modelos de diseño no se encuentran completos, los detalles se refinan durante el proceso de codificación
3. Analizar detenidamente el ambiente de implementación
4. Los diseñadores también deben codificar. Cuando el modelo es desarrollado por alguien más se entrega a otra persona para codificarlo hay un riesgo significativo de que no capte sus detalles adecuadamente. Separar las funciones del diseño y la codificación es riesgosa, es mejor contar con especialistas en el equipo que puedan diseñar y codificar.
5. Realizar análisis por riesgos.

De estas cinco prácticas, la 1 y la 3 son realizadas en un 100% en las empresas consultadas.

No se presenta ninguna de las 5 prácticas que alguna empresa no realice lo que muestra sintonía entre las buenas prácticas usadas a nivel internacional y con la industria nacional.

La práctica 2 y la 4: “Los modelos de diseño no se encuentran completos, los detalles se refinan durante el proceso de codificación”.

“Los diseñadores también deben codificar. Cuando el modelo es desarrollado por alguien más se entrega a otra persona para codificarlo hay un riesgo significativo de que no capte sus detalles adecuadamente. Separar las funciones del diseño y la codificación es riesgosa, es mejor contar con especialistas en el equipo que puedan diseñar y codificar” (Velázquez, 2012).

Ambas prácticas están relacionadas con la importancia de llevar de la mano el diseño y la codificación, en lo que están de acuerdo el 83% con SIEMPRE y el 17% con A VECES. Esta es la razón por la cual los jóvenes deben conocer de diseño.

Otros aspectos importantes que mencionan los empresarios son la revisión entre pares, realizar una entrega formal de diseñador a programador y realizar reuniones por grupos de trabajo como una manera de conocer el proyecto en su totalidad.

En cuanto a la práctica 5: “Realizar análisis por riesgos” puede decirse que el 66% de las empresas usan esta práctica SIEMPRE y el restante 34% la usan AVECES, lo que indica falta de una cultura en el campo de la prevención para la realización de software.

8.1.5.3 Etapa Desarrollo

Se presentaron 7 prácticas:

1. Probar con código. Nunca debe asumirse que un diseño funciona, sino que debe probarse codificándolo para determinar si funciona.
2. La retroalimentación es importante. Nunca debe olvidarse que es necesario buscar activamente retroalimentación sobre el trabajo que se realiza. Esto permite mejorar el sistema
3. Utilizar herramientas de generación de código.
4. Documentar las partes complicadas del sistema. A través de la documentación generada debemos poder entender el funcionamiento del sistema, así como las razones que sustentan las decisiones de diseño
5. Verificación y seguimiento de estándares
6. Adecuada planificación de la construcción y mantener mediciones acerca del desempeño y calidad de los programadores

7. Usar con propiedad los diferentes lenguajes que apoyan el proceso de construcción en los diferentes niveles de la arquitectura: lenguajes de presentación, lenguajes de programación, lenguajes de configuración, lenguajes de bases de datos.

Estas 7 prácticas se realizan SIEMPRE por las empresas entrevistadas, la información correspondiente a NO APLICA está relacionada con otra empresa que realiza aseguramiento de la calidad, capacitación y entrenamiento, reservándose el derecho a contestar.

Como prácticas asociadas identificadas por los entrevistados está la Entrega al cliente por parte de los programadores como un reconocimiento a su labor y la Revisión por herramientas desde arquitectura y otras de análisis estático de código, como: Sonar (www.sonarqube.org/), PMD (pmd.sourceforge.net/), Jenkins (jenkins-ci.org/) que pueden ser útiles para la verificación del trabajo realizado.

8.2 Fase 2: Mejores prácticas de calidad de desarrollo de software para la media técnica en programación

El propósito es seleccionar un conjunto de mejores prácticas para el desarrollo de software que los estudiantes de la media técnica deben implementar en su formación, cumpliendo las competencias que deben desarrollar, al ser comparadas con las implementadas en la industria. Los Cuadros 11 a 14 muestran las prácticas identificadas en las empresas consultadas y las que se consideran deben ser integradas en la formación de media técnica.

El criterio para definir la prioridad de implementación de las buenas prácticas identificadas en la fase I, consiste en empezar por aquellas que se implementan con mayor frecuencia en las empresas, como se observa en la columna “cumplidas siempre en las empresas seleccionadas”

Cuadro 11 Selección de prácticas Organizacionales (Construcción Propia)

Prácticas Organizacionales		
Identificadas y presentadas en la encuesta a los empresarios	Cumplidas Siempre en las empresas consultadas	Seleccionadas para la media técnica
Revisar el trabajo de otros en forma objetiva, sincera y propiamente documentada	86%	X
Escuchar equitativamente las opiniones, preocupaciones y quejas de un colega	72%	X
Ayudar a sus colegas a que estén totalmente alertas a los actuales estándares incluyendo políticas y procedimientos de protección de contraseñas, archivos, información confidencial y las medidas de seguridad en general.	86%	Es un proceso elevado para un nivel de la técnica
En situaciones fuera de sus propias áreas de competencia, solicitar las opiniones de otros profesionales que tengan competencia en esa área.	72%	X
Mejorar su conocimiento de los avances en el análisis, especificación, diseño, desarrollo, mantenimiento, pruebas del software y documentos relacionados, junto con la administración del proceso de desarrollo	72%	X
Mejorar su habilidad para producir documentación precisa, informativa y bien redactada.	100%	X
Mejorar su comprensión del software de los documentos con que se trabaja y del medio ambiente donde serán usados	100%	X
Calidad en el proceso dará calidad en el producto	100%	X
Asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)	100%	X
Uso de arquitectura basada en componentes	100%	X
Control de cambios	100%	X
Modelado visual del software	100%	X
El aseguramiento de la calidad es parte del proceso de desarrollo y no la responsabilidad de un grupo independiente	100%	X
Hay cultura organizacional, equipos auto-organizados, auto – control, auto –capacitación	100%	X
Documentar lecciones aprendidas	100%	X
Se realiza revisión entre pares	100%	X

Cuadro 12 Selección prácticas de requisitos (Construcción Propia)

Practicas Etapa Requisitos		
Identificadas y presentadas en la encuesta a los empresarios	Cumplidas Siempre en las empresas consultadas	Seleccionadas para la media técnica
Identificar, definir y atender asuntos éticos, económicos, culturales, legales y ambientales relacionados a los proyectos de trabajo.	66.6%	X
Participación activa de los clientes	100%	X
Visualización de los requerimientos organizados por prioridad.	100%	X
Comunicación asertiva con su equipo de trabajo, entrega de información de una sola fuente.	100%	X
Visualización de la arquitectura, primero desde un nivel de abstracción alto para identificar una estrategia que permita la implementación de la solución.	100%	X
Elaborar documentación entregable a través del ciclo de vida del producto de forma paralela a la creación de la solución	83%	X
Ver más allá del modelado. Algunas veces los requerimientos que tienen una mayor prioridad son complejos, lo cual implica explorarlos antes de comenzar su desarrollo para reducir el riesgo general del desarrollo.	50%	X
(Nuevas) Verificación que no se pasó algo por alto y Visita al cliente	1%	X

Cuadro 13 Selección prácticas análisis – diseño (Construcción Propia)

Practicas Etapa Análisis y Diseño		
Identificadas y presentadas en la encuesta a los empresarios	Cumplidas Siempre en las empresas consultadas	Seleccionadas para la media técnica
El diseño general del sistema se construye conforme avanza el desarrollo del proyecto cambiando y evolucionando constantemente	100%	X
Los modelos de diseño no se encuentran completos, los detalles se refinan durante el proceso de codificación.	83%	X
Analizar detenidamente el ambiente de implementación.	100%	X
Los diseñadores también deben codificar.. Cuando el modelo es desarrollado por alguien más se entrega a otra persona para codificarlo hay un riesgo significativo de que no capte sus detalles adecuadamente. Separar las funciones del diseño y la codificación es riesgosa, es mejor contar con especialistas en el equipo que puedan diseñar y codificar.	83%	X
Realizar análisis por riesgos	66%	X

Cuadro 14 Selección prácticas desarrollo (Construcción Propia)

Prácticas Etapa Desarrollo		
Identificadas y presentadas en la encuesta a los empresarios	Cumplidas Siempre en las empresas consultadas	Seleccionadas para la media técnica
Probar con código. Nunca debe asumirse que un diseño funciona, sino que debe probarse codificándolo para determinar si funciona	100%	X
La retroalimentación es importante. Nunca debe olvidarse que es necesario buscar activamente retroalimentación sobre el trabajo que se realiza. Esto permite mejorar el sistema.	100%	X
Utilizar herramientas de generación de código.	100%	No se asume esta práctica pues los estudiantes deben conocer el código. En esta etapa inicial no es conveniente para su proceso de formación. Mientras que el mundo de la industria debe ahorrarse tiempo.
Documentar las partes complicadas del sistema. A través de la documentación generada debemos poder entender el funcionamiento del sistema, así como las razones que sustentan las decisiones de diseño.	100%	X
Verificación y seguimiento de estándares	100%	X
Adecuada planificación de la construcción y mantener mediciones acerca del desempeño y calidad de los programadores	100%	X
Usar con propiedad los diferentes lenguajes que apoyan el proceso de construcción en los diferentes niveles de la arquitectura: lenguajes de presentación, lenguajes de programación, lenguajes de configuración, lenguajes de bases de datos	100%	X

8.3 Fase 3. Propuesta para la implementación de mejores prácticas de desarrollo de software en el proceso formativo de los estudiantes de media técnica

La propuesta de instrumentos, está enfocada a la calidad en el proceso de software, relacionada con las personas; esto quiere decir, que los estudiantes de la media técnica desde el inicio de su formación, serán conscientes de lo que implica la calidad, siendo más competitivos a la hora de enfrentar el mundo real.

La propuesta busca satisfacer principios básicos del aprendizaje como:

- Es más fácil aprender desde la formación inicial las buenas prácticas de calidad, que luego pretender cambiar esquemas arraigados.
- Favorecer las revisiones continuas, retroalimentación constante, a la entrega de software que agregue valor, entregas periódicas en poco tiempo.
- Permitir la estandarización de prácticas que permitan seguir un proceso por cada integrante del equipo del proyecto pedagógico integrador (PPI).
- Posibilitar el ingreso de personas, en diferentes momentos del proceso, ayudando a valorar y hacer ajustes fácilmente al mismo.
- Servir de oportunidad para la mejora continua en los procesos de aprendizaje.

Es importante destacar, que los estudiantes desde el comienzo del proyecto en grado 10°, se enfrentan al conocimiento de la necesidad o problema de la empresa, y realizan los procesos de análisis, diseño y desarrollo, por lo tanto, la propuesta

pretende que se simule un proceso empresarial donde se viven todas las etapas de un proyecto, y el estudiante vaya aprendiendo a partir de espacios lúdicos que vivan la experiencia de conocerlas.

Dentro de las estrategias que ha implementado la Alianza Futuro Digital está el Laboratorio didáctico del software, ubicado dentro de las instalaciones del Vivero del Software, en el Politécnico Jaime Isaza Cadavid y el ITM, como espacio diseñado para implementar por parte de los estudiantes las mejores prácticas para las fases de Análisis, Desarrollo, Pruebas e Implantación; esta propuesta tendría una excelente posibilidad de desarrollarse allí.

8.3.1 Instrumentos para la implementación de mejores prácticas de desarrollo de software en la media técnica en programación.

El instrumento presentado es el fruto de las buenas prácticas identificadas que usan hoy en la industria del software, aplicables al proceso de formación de los estudiantes en la educación media, técnica y tecnológica; dichas prácticas funcionan e imprimen un sello de calidad al proceso y por tanto al producto. Este instrumento representa una guía, para que el maestro/orientador monitoree el proceso de desarrollo, y promueva una autoevaluación para los equipos de trabajo dirigido a todos y cada uno de los integrantes, como parte del seguimiento al proceso y con el fin de tomar acciones para la mejora. El instrumento también podrá ser usado para realizar verificaciones en cualquier momento, o al momento de realizar entregas.

Por otro lado, se incluyen también en el instrumento, prácticas fruto de la revisión detallada de las normas que aplican a la calidad en cuanto al proceso. Las buenas prácticas fueron además clasificadas, de acuerdo a lo organizacional, etapa de requisitos, análisis - diseño y desarrollo.

El instrumento presentado consiste en una rúbrica para chequear y valorar las prácticas a implementar en el proceso formativo de los estudiantes. A partir del Cuadro 15a, se presenta la rúbrica que permite asignar una valoración de 1 a 4 a cada práctica, según su estado de implementación por parte de los estudiantes, así:

Práctica no implementada (1 punto): Corresponde a que dicha práctica no ha sido asumida por los estudiantes.

Práctica implementada en forma básica pero sin persistencia (2 puntos): La práctica ha sido incorporada en el trabajo de los estudiantes, pero no como práctica cotidiana.

Práctica implementada efectivamente (3 puntos): La práctica ha sido incorporada en el trabajo de los estudiantes, como algo cotidiano.

Práctica implementada alcanza los objetivos de los procesos (4 puntos): La práctica ha sido incorporada en el trabajo de los estudiantes, reconociéndola como un medio eficaz para lograr los objetivos.

La propuesta con estas prácticas, no es considerarlas como una actividad más que hacer, sino incorporarlas en el día a día de la formación, con el fin de que los estudiantes adquirieran la cultura de su implementación.

La aplicación de la rúbrica permitirá al totalizar los puntajes, determinar las falencias a trabajar y fortalecer las buenas prácticas, siguiendo las estrategias y formas de implementación.

Cuadro 15a Instrumento verificación prácticas organizacionales (Construcción Propia)

Módulos que tienen compromiso con el desarrollo de estas buenas prácticas organizacionales: < Identificación del ciclo de vida del software < Habilidades comunicativas y de negociación < Construcción de informes empleando herramientas ofimáticas < Identificar las organizaciones, sistemas de información empresarial			Nivel de logro			
N°	Descriptor práctica Organizacional	Aclaraciones para su implementación	Práctica no implementada (1 punto)	Práctica implementada en forma básica sin persistencia (2 puntos)	Práctica implementada efectivamente (3 puntos)	Práctica implementada alcanza los objetivos de los procesos (4 puntos)
1	El equipo debe conocer y aceptar los objetivos del proyecto	Conocer las condiciones del desarrollo del proyecto software				
2	El equipo esta auto organizado	Los estudiantes se organizaron como equipo y son conscientes de la implicación del trabajo en equipo El líder debe fortalecer competencias de los estudiantes de liderazgo, manejo de grupos, habilidad para tomar decisiones, capacidad para resolver problemas difusos, generar confianza y respeto mutuos.				
3	El equipo debe conocer y aceptar las tareas necesarias para completar cada requisito	El equipo debe explicar con sus palabras lo que debe hacer				
4	Estimar el esfuerzo necesario para realizar cada tarea.	De acuerdo a la experiencia debe repartirse tareas teniendo en cuenta las habilidades de los participantes y su agilidad para el desarrollo				
5	Cada miembro del equipo se auto asigna a las tareas.	Comprometerse con lo que quiero hacer Equipos disciplinados con total autonomía				
6	Cada miembro del equipo tiene orientación al logro.	El proyecto sale adelante si cada miembro del equipo hace todo lo que más puede para lograr el objetivo				
7	Conocer las habilidades de cada miembro del equipo para identificar su aporte	Aplicar test que permitan identificar aptitudes, habilidades (Inteligencias múltiples)				

Cuadro 15b Instrumento verificación prácticas organizacionales (Construcción Propia)

N°	Descriptor práctica Organizacional	Aclaraciones para su implementación	Nivel de logro			
			Práctica no implementada (1 punto)	Práctica implementada en forma básica sin persistencia (2 puntos)	Práctica implementada efectivamente (3 puntos)	Práctica implementada alcanza los objetivos de los procesos (4 puntos)
8	Dedicación de tiempo en sus tareas por parte de cada miembro	Reuniones de verificación de tareas				
9	Reuniones diarias de seguimiento de equipo no mayores a 15 minutos y preguntarse ¿Que hice ayer, Que haré hoy y que dificultades tengo	Estrategias de liderazgo y manejo de grupos, que permitan resolver dificultades y seguir adelante.				
10	Destinación de reuniones de retroalimentación con el líder proyecto(docente)	Usar un diagrama que permita ver la relación tiempos y tareas. Colocarlo en un sitio visible				
11	Revisar el trabajo de otros en forma objetiva, sincera y propiamente documentada.	Generar discusiones para analizar situaciones				
12	Escuchar equitativamente las opiniones, preocupaciones y quejas de un colega.	Generar lluvia de ideas para resolver problemas				
13	En situaciones fuera de sus propias áreas de competencia, solicitar las opiniones de otros profesionales que tengan competencia en esa área.	Gestionar para solicitar ayuda				
14	Mejorar su habilidad para producir documentación precisa, informativa y bien redactada.	Cuando alguien lee se podrá verificar que tan claro quedó lo escrito				
15	Se realiza control de cambios	Conocer y usar políticas de control de versiones con el fin de no usar información obsoleta				
16	Se realiza modelado visual del software, documento escrito y auditivo que permita la comprensión de la información	Crear diagramas, líneas del tiempo, mapas mentales, de ideas conceptuales y podcast que permitan comprender el proyecto				
17	Se realiza revisión entre pares	Generar reuniones para compartir , revisar documentos y errores				

Cuadro 15c Instrumento verificación prácticas organizacionales (Construcción Propia)

N°	Descriptor práctica Organizacional	Aclaraciones para su implementación	Nivel de logro			
			Práctica no implementada (1 punto)	Práctica implementada en forma básica sin persistencia (2 puntos)	Práctica implementada efectivamente (3 puntos)	Práctica implementada alcanza los objetivos de los procesos (4 puntos)
18	Documentar lecciones aprendidas	Se hace reflexión continua frente a lo bueno, lo repetible y no repetible. Tener en cuenta la experiencia vivida en otros proyectos ayuda a la estimación y al desarrollo del proyecto.				
19	El aseguramiento de la calidad es parte del proceso de desarrollo y no la responsabilidad de un grupo independiente	Todos aportamos a la calidad del producto cuando hacemos las actividades que nos corresponden bien y es valorada cada acción. Trabajar bien desde el principio, es siempre la forma más rápida y económica. (Calidad en el proceso dará calidad en el producto)				
20	Hay cultura organizacional, equipos auto-organizados, auto – control, auto – capacitación	Generar consciencia de lo importante que es auto - dirigirse. Usar videos de automotivación y experiencias replicables.				
21	Comunicación con retroalimentación para corroborar la información que se está pasando.	Comunicación asertiva con su equipo de trabajo, entrega de información de una sola fuente.				
22	Es más importante hoy las personas y sus interacciones que las herramientas y los procesos.	Las personas con su creatividad están en capacidad de resolver cualquier situación				
23	Capacidad de resolución de diferencias entre las personas para llegar a acuerdos en el desarrollo del trabajo	Usar estrategias para la resolución de conflictos y diferencias. En el grupo prima la responsabilidad con los compromisos individuales.				
24	Monitorear y controlar el proceso	Generar cultura de mejora de procesos				
25	Buena planeación	Definir el plan de trabajo con actividades específicas y tiempos				

N°	Descriptor práctica Organizacional	Aclaraciones para su implementación	Nivel de logro			
			Práctica no implementada (1 punto)	Práctica implementada en forma básica sin persistencia (2 puntos)	Práctica implementada efectivamente (3 puntos)	Práctica implementada alcanza los objetivos de los procesos (4 puntos)
26	Mejorar su conocimiento de los avances en el análisis, especificación, diseño, desarrollo, mantenimiento, pruebas del software y documentos relacionados, junto con la administración del proceso de desarrollo	La información acerca del proceso debe fluir entre los integrantes del equipo con el fin de conocer los avances, dificultades y plantear alternativas de solución.				
27	Mejorar su comprensión del software de los documentos con que se trabaja y del medio ambiente donde serán usados	Debe avanzarse cada vez en conocer el proceso				
28	Uso de arquitectura basada en componentes	Partir el problema en partes será más fácil para resolverlo				
29	Para facilitar un mejor ensamblaje de los procesos, lograr procesos comunicativos que afiancen la coherencia entre las personas.	La comunicación es clave para evitar re procesos, pérdida de tiempo y dificultades interpersonales.				
30	Realizar análisis por riesgos	Generar cultura de gestión de riesgos, con el fin de prevenir situaciones que puedan afectar el proceso. Tomar riesgos razonables				
31	Verificación que no se pasó algo por alto	Siempre debe revisarse y entre varias personas idóneas con el fin de no dejar pasar asuntos importantes.				

Cuadro 16 Instrumento verificación prácticas requisitos (Construcción Propia)

Módulos que tienen compromiso con el desarrollo de estas buenas prácticas en requisitos:			Nivel de logro			
<ul style="list-style-type: none"> Identificación del ciclo de vida del software Habilidades comunicativas y de negociación Construcción de informes empleando herramientas ofimáticas Interpretación de requerimientos Identificación de estándares para la documentación Identificar las organizaciones, sistemas de información empresarial 						
N°	Descriptor práctica de Requisitos	Aclaraciones para su implementación	Práctica no implementada (1 punto)	Práctica implementada en forma básica sin persistencia (2 puntos)	Práctica implementada efectivamente (3 puntos)	Práctica implementada alcanza los objetivos de los procesos (4 puntos)
1	Conocer el contexto del problema y sus características. Transformar las necesidades de las partes interesadas en requisitos de cliente y/o historias de usuario.	Realización de reuniones y entrevistas (preliminares y específicas) que son documentadas, a través de Historias de usuario: Como [rol] necesito o requiero [descripción de la funcionalidad] con la finalidad de [Razón o resultado] con el fin de comprender los requisitos del cliente				
2	Identificar, definir y atender asuntos éticos, económicos, culturales, legales y ambientales relacionados a los proyectos de trabajo.	Comprender los contextos del problema, empleando mapas conceptuales y/o mentales, que permitan reconocer la realidad del medio.				
3	Gestión de los requerimientos que permita encontrar un equilibrio que satisfaga los deseos de todos Visualización de los requerimientos organizados por prioridad.	Una vez hay claridad en los requisitos, de acuerdo a las necesidades del cliente, se asignan pesos (de acuerdo al grado de dificultad) y a la vez dar priorización a su desarrollo. (Planning poker)				
4	Participación activa de los clientes	Constante conversación entre cliente – estudiante – docente articulado y articulador con el fin de acordar el alcance del proyecto.				
5	Visualización de la arquitectura, primero desde un nivel de abstracción alto para identificar una estrategia que permita la implementación de la solución.	Cruzar la información relacionada con el alcance, los pesos asignados a los requisitos para tener una idea macro de la solución.				
6	Visualización de requerimientos complejos en partes más pequeñas Los requerimientos que tienen una mayor prioridad son complejos, lo cual implica explorarlos antes de comenzar su desarrollo para reducir el riesgo general del desarrollo	Comprender las partes que componen el alcance, manteniendo las historias de usuarios de forma simple.				
7	Elaborar documentación entregable a través del ciclo de vida del producto de forma paralela a la creación de la solución	El alcance y los requisitos priorizados expuestos en un tablero de tareas con responsables asignados son una estrategia para llevar una documentación visual y útil a este nivel.				
8	Involucrar el equipo testing desde el principio del proceso	Hacer consciente al (los) estudiante(s) que tiene(n) rol de tester, la importancia de prever las pruebas que permitirán probar el correcto funcionamiento del sistema.				
9	Ser consciente que la calidad del proceso está en esta etapa	La comprensión del problema por parte de los involucrados hace que se brinde alternativas de solución.				

Cuadro 17 Instrumento verificación prácticas análisis - diseño (Construcción Propia)

Módulos que tienen compromiso con el desarrollo de estas buenas prácticas de Análisis - Diseño:			Nivel de logro			
<ul style="list-style-type: none"> Identificación del ciclo de vida del software Habilidades comunicativas y de negociación Construcción de informes empleando herramientas ofimáticas Interpretación de requerimientos Identificación de estándares para la documentación Identificar las organizaciones, sistemas de información empresarial Desarrollo pensamiento analítico y sistémico Construcción de Base de datos Construcción de elementos de software I Construcción de elementos de software web 						
Nº	Descriptor práctica de Análisis - Diseño	Aclaraciones para su implementación	Práctica no implementada (1 punto)	Práctica implementada en forma básica sin persistencia (2 puntos)	Práctica implementada efectivamente (3 puntos)	Práctica implementada alcanza los objetivos de los procesos (4 puntos)
1	Comprender bien el concepto de arquitectura por capas.	Es fundamental comprender el cómo se realizará la programación del sistema, las ventajas de separar la lógica del negocio de la de diseño.				
2	Alto nivel de abstracción para favorecer discusiones sobre diversos niveles y soluciones arquitectónicas para la solución	Este nivel de abstracción es posible cuando el estudiante conoce alternativas, a través de ejemplos, estudios de casos y esquemas que interpreta y construye.				
3	El diseño general del sistema se construye conforme avanza el desarrollo del proyecto cambiando y evolucionando constantemente	Como trabajo en equipo que es el desarrollo de software, distintas visiones y experiencias de los participantes es posible que se generen cambios y evolución, que representan también una posibilidad de aprendizaje.				
4	Los modelos de diseño no se encuentran completos, los detalles se refinan durante el proceso de codificación.					
5	Los estudiantes deben dibujar las interfaces siguiendo una lista de chequeo de elementos de diseño.	El uso de una lista de chequeo permite verificar si se están teniendo en cuenta todos los aspectos necesarios. Esta parte gráfica permite identificar el criterio de usabilidad. Debe hacerse el ejercicio de cruzar el análisis de la solución y la forma como lo verá el usuario.				
6	Identificar los requisitos de interfaz.					
7	Analizar y validar los requisitos.	No dar por sentado nada, hacer retroalimentación con el cliente, para validar la información.				
8	Establecer los conceptos y los escenarios de operación					
9	Establecer una definición de la funcionalidad y de los atributos de calidad requeridos.	Todo debe estar documentado y comunicado para brindar la información a todos. De ahí un buen tablero de tareas práctico y visible				
10	Analizar detenidamente el ambiente de implementación.	Esto permitirá conocer todas las partes del proceso, importante para tener una visión de lo que se viene.				
11	Los diseñadores también deben codificar.	Es conocer el campo del otro, como se				
12	Cuando el modelo es desarrollado por alguien más se entrega a otra persona para codificarlo hay un riesgo significativo de que no capte sus detalles adecuadamente. Separar las funciones del diseño y la codificación es riesgosa, es mejor contar con especialistas en el equipo que puedan diseñar y codificar.	lleva a cabo la estructura dada en el pensamiento o en papel ya en el lenguaje de programación. Esto explica la importancia de conocer las tareas detalladas de todos los roles.				

Cuadro 18a Instrumento verificación prácticas de Desarrollo (Construcción Propia)

Módulos que tienen compromiso con el desarrollo de estas buenas prácticas de Desarrollo:			Nivel de logro			
<ul style="list-style-type: none"> • Identificación del ciclo de vida del software • Habilidades comunicativas y de negociación • Construcción de informes empleando herramientas ofimáticas • Interpretación de requerimientos • Identificación de estándares para la documentación • Identificar las organizaciones, sistemas de información empresarial • Desarrollo pensamiento analítico y sistémico • Construcción de Base de datos • Construcción de elementos de software I • Construcción de elementos de software web 						
Nº	Descriptor práctica de Desarrollo	Aclaraciones para su implementación	Práctica no implementada (1 punto)	Práctica implementada en forma básica sin persistencia (2 puntos)	Práctica implementada efectivamente (3 puntos)	Práctica implementada alcanza los objetivos de los procesos (4 puntos)
1	Revisión de pares en documentos de identificación de errores	Documentar los errores servirá de insumo para resolverlos en el futuro, además de hacer una revisión colaborativa.				
2	Incentivar la realización de Pruebas unitarias(probarse así mismo)	Una vez se tiene un mejor dominio del uso del lenguaje de programación, es posible empezar a probar su código, de ahí la importancia de las pruebas de escritorio.				
3	Interacción con el cliente en entregas tempranas	El cliente como conocedor de su línea de trabajo, debe validar los avances.				
4	Probar con código. Nunca debe asumirse que un diseño funciona, sino que debe probarse codificándolo para determinar si funciona	Revisar si lo planteado en diseño funciona bien.				
5	Buscar activamente retroalimentación sobre el trabajo que se realiza.	La comunicación fluida dentro del equipo es sumamente importante para compartir ideas, hacer preguntas y aprender.				
6	Documentar las partes complicadas del sistema. A través de la documentación generada debemos poder entender el funcionamiento del sistema, así como las razones que sustentan las decisiones de diseño.	La documentación como una forma de llevar registro de las lecciones aprendidas.				
7	Verificación y seguimiento de estándares de codificación	Hacer esta lista de chequeo, posibilitará afianzamiento en los estudiantes respecto al manejo de la documentación.				
8	Adecuada planificación de la construcción y mantener mediciones acerca del desempeño y calidad de los programadores	La buena planificación, acompañada de la verificación de la eficacia de la comunicación son claves para un buen proceso en el desarrollo. Con el fin de resolver dificultades y avanzar en la formación de los estudiantes.				
9	Usar con propiedad los diferentes lenguajes que apoyan el proceso de construcción en los diferentes niveles de la arquitectura: lenguajes de presentación, lenguajes de programación, lenguajes de configuración, lenguajes de bases de datos	Afianzar la competencia del uso de lenguajes, y su estudio de forma autodidacta, para adquirir competencias en resolución de problemas.				
10	Contabilizar el tiempo que se invierte para desarrollar determinada funcionalidad permitirá tener claridad de ese tiempo para desarrollar otra funcionalidad.	Esto permite conocerse, ir teniendo su propio banco de código y aprender a estimar en cuanto a tiempo.				

Cuadro 18b Instrumento verificación prácticas de Desarrollo (Construcción Propia)

Nº	Descriptor práctica de Desarrollo	Aclaraciones para su implementación	Práctica no implementada (1 punto)	Práctica implementada en forma básica sin persistencia (2 puntos)	Práctica implementada efectivamente (3 puntos)	Práctica implementada alcanza los objetivos de los procesos (4 puntos)
11	Los defectos deben registrarse tan pronto se identifican	Esta práctica permite no dejar pasar ningún detalle importante, que podría				
		afectar la confiabilidad en el producto software. Identificar fecha, descripción del defecto, en qué fase se detectó, cuando se solucionó.				
12	Programar bien con buenas prácticas y con expectativas de aprender	Esto garantiza competencias en los estudiantes, técnica, tiempo y dinero, para todos.				

9 CONCLUSIONES

El desarrollo de software es una actividad compleja, que requiere la integración de factores técnicos, gerenciales y organizacionales. De ahí que el reto en la formación de los estudiantes esté orientada a mejorar la práctica del desarrollo en las organizaciones de software. El papel de los docentes radica en tener una vista unificada acerca del proceso de desarrollo de software, percepción de la realidad de esta práctica en las organizaciones de software, conocer los problemas que dichas organizaciones enfrentan debido a la falta de aplicación de buenas prácticas de ingeniería de software, y el conocimiento de los estándares internacionales para poder apuntar a la competitividad. Las estrategias identificadas y propuestas están encaminadas a formar las competencias requeridas, de tal forma que los futuros desarrolladores estén convencidos de que las prácticas en ingeniería de software son claves para el desarrollo exitoso del software.

No existe en forma absoluta, una metodología para desarrollo de software mejor que otra, sino que ésta, debe estar alineada a la cultura de la organización, tamaño y complejidad del proyecto, así como al ciclo de vida del producto que pretende desarrollarse; de ahí tendrán que tenerse en cuenta varias variables, una es que las empresas cuando buscan certificaciones de calidad para el proceso y el producto, desean entre muchas cosas organizar procedimientos, pero esto es significativo en la medida que exista una cultura organizacional que permita tomar conciencia por parte del equipo de trabajo.

No fue fácil recopilar las buenas prácticas, categorizarlas y lograr comprender la intención de los entrevistados para lograr la fidelidad de la información, lo importante es que se escuchó a los empresarios sobre su visión del proceso de formación de la media técnica hasta ahora, y ver viable la incorporación prácticas de calidad para el desarrollo de software, al interior de la formación de los jóvenes de la media técnica en programación de las instituciones educativas de la Alianza Futuro Digital, ya que

el proceso que se ha venido realizando durante estos 9 años, tiene bases en cuanto al currículo, la metodología y las metas a alcanzar.

Fue posible identificar la diferencia entre la calidad enfocada en el proceso y en el producto, para centrarse en la primera que es la relacionada con buenas prácticas y con personas, entendidas como aquellas acciones, donde se ha dado coincidencia de que hay prácticas que son exitosas en determinados contextos y que es posible replicarlas en el ámbito educativo.

Incorporar las buenas prácticas en la formación de la media técnica, en el día a día en el desarrollo del Proyecto Pedagógico Integrador (PPI) es posible y valioso, con el fin de pensar distinto el desarrollo de software, en el cual un óptimo proceso permitirá un óptimo producto.

El Proyecto Pedagógico Integrador, como simulación de un proyecto real de software, permite que los estudiantes entiendan y apliquen el proceso de levantamiento de requerimientos a partir de un problema real, realicen el análisis y el diseño, luego programen y realicen pruebas unitarias. Desarrolla además, competencias organizacionales importantes para conocer el contexto, y desarrolla la habilidad del trabajo en equipo y del liderazgo, al trabajar con otros compañeros, asumiendo roles y enfrentando las situaciones cotidianas de la convivencia.

Estos aprendizajes, lo ratifican los empresarios cuando destacan el buen desarrollo con que vienen los estudiantes en cuanto al trabajo en equipo, orientación al logro y manejo de TIC. El reto, queda en fortalecer desde la formación básica el pensamiento algorítmico, analítico y sistémico, y la comprensión organizacional de cómo funcionan estos procesos.

Las actividades de toma de requisitos, análisis y diseño, antes destinadas únicamente a los ingenieros, cuentan hoy en día con la participación de técnicos y tecnólogos en un 28.5% de la muestra tomada, lo que indica que la formación de los

estudiantes relacionada con el trabajo de todas las etapas del proceso de desarrollo de software, ha sido significativo y valioso a la hora de enfrentarse al mundo del trabajo.

Llevar a cabo la propuesta planteada con este proyecto, implica tener el convencimiento que funciona, empezar a implementarla para evaluar como lo hace, y comprometer la voluntad de los docentes en incentivar la incorporación de estas buenas prácticas.

El talento humano estudiante y maestros es clave para lograr que los procesos funcionen bien, exista motivación y comprensión.

Se encontraron una buena cantidad de prácticas organizacionales sobre las específicas relacionadas con las etapas de desarrollo de software, la empresa está dispuesta incluso, a fortalecerlas si es necesario, con tal de que el estudiante posea bases firmes en las competencias básicas para su rol en el proceso de software.

El instrumento desarrollado permite valorar de 1 a 4 el nivel que se tiene en la adquisición de la práctica, lo que posibilita afianzar el logro hasta llegar a la valoración 4 y crear cultura una vez se llega. Este proceso, logra ser transparente para cada estudiante, los docentes y los empresarios en la medida en que haga parte de la cultura.

La experiencia adquirida en la media técnica, el conocimiento de los estudiantes, la forma como aprenden, hizo posible adecuar las prácticas al desarrollo en el aula, permitiendo una formación en cultura de lecciones aprendidas, para un mejor desempeño que implica:

- Trabajar más sobre la prevención a través de la Gestión del Riesgo.
- Equipos integrados para conocer los avances de las etapas del proyecto.
- Representan la cultura de la mejora continua que permite a los miembros de la organización pensarse como parte de un engranaje que cada vez logra mejores resultados.

10 TRABAJOS FUTUROS

Uno de los temas que podrían partir de esta investigación es lo relacionado con las prácticas de pruebas que durante este ciclo de formación (media técnica) se tocan someramente y que por tal motivo no se tuvieron en cuenta.

Validar el instrumento planteado, con el fin de revisar los aportes que pueda hacer al proceso de formación de los estudiantes.

11 BIBLIOGRAFIA

Abad Londoño, J. H. E. (2012). Framework para el desarrollo de software en entornos académicos. Universidad Eafit.

Abad Londoño, J. H. S. H. (2013). Scrum: Cediendo el mando y control AL EQUIPO. Recuperado de: <http://www.intersoftware.org.co/content/scrum-cediendo-el-mando-y-control-al-equipo>.

Abrahamsson, P., & Warsta, J. (2003). New Directions on Agile Methods: A Comparative Analysis. Proceedings of the International Conference on Software Engineering.

Agudelo Velásquez, O. L. F. D.-M. (2013). Alianza Futuro Digital Medellín: Una experiencia de articulación y formación de talento humano para el sector del Software, desde la educación media hasta la tecnológica. XV Encuentro Virtual Educa.

Albaladejo, X. (2014). ¿Proyectos complejos? ¿Necesitas resultados? Conoce Scrum. Recuperado de: <http://www.proyectosagiles.org/>

Anaya, R. E. (2006). Una visión de la enseñanza de la Ingeniería de Software como apoyo al mejoramiento de las empresas de software, 42, N 141.

Arboleda, H y otros. (2009) Product derivation in a Model- Driven software product line using decision models. University of Los Andes. Colombia

Beck, Kent, et al. (2001). "Manifiesto for Agile Software Development". Agile Alliance. Recuperado el 21 de junio de 2012 de: <http://agilemanifesto.org/iso/es/>

Bruegge y Dutoit. (2002) "Ingeniería de Software Orientado a Objetos". Prentice Hall.

Bernhart, M., Grechenig, T., Hetzl, J., & Zuser, W. (2006). Dimensions of software engineering course design. Proceeding of the 28th international conference on Software engineering - ICSE '06,667. doi:10.1145/1134285.1134387

Bocanegra, J y otros. (2009). Modelado de negocio interorganizacional: una aproximación para la trazabilidad entre objetivos, modelos organizacionales y procesos de negocio. Facultad de Ingeniería. Universidad de la Amazonía. Florencia – Colombia.

Builes V. C. A. (2008) "Notas acerca de Ingeniería de Software" Ed. Artes y Letras Ltda. Colombia.

Cardona T. J. A. del Q. (2012). Diseño de una estrategia de aprendizaje para implementar prácticas de PSP y TSP en cursos básicos de programación. Caso programa de Ingeniería de Sistemas y computación Universidad del Quindío. Universidad Eafit.

Cataldi, et al., (s.f.). Ingeniería de software educativo. Recuperado el 21 de octubre de 2015, de <http://laboratorios.fi.uba.ar/lsi/c-icie99-ingenieriasoftwareeducativo.pdf>.

Caro, A., & Fuentes, A. (2013). Developing information systems focused on data quality. *Ingeniare. Revista Chilena de Ingeniería*, 21, N 1, 54–69.

Cendejas, J., (2014). Implementación del modelo integral colaborativo (mdsic) como fuente de innovación para el desarrollo ágil de software en las empresas de la

zona centro - occidente en México. Universidad Popular Autónoma del Estado de Puebla.

Centro nacional de tecnologías de información. (2012) Metodología de la red nacional de integración y desarrollo de software libre (MeRinde): Guía detallada. Recuperado el 16 de Junio de 2012, de :<http://merinde.net>.

Chrissis, et al (2007) CMMI: Guidelines for Process

Chua, B. B., & Dyson, L. E. de tecnología de S. (2004). La aplicación del modelo ISO 9126 para la evaluación de un sistema de e-aprendizaje. Recuperado de: <http://www.ascilite.org.au/conferences/perth04/procs/chua.html>

Dellacanáonica, O (2008) Best Practices en sincronía con cada organización Recuperado el 21 de octubre de 2015, de:<http://www.evaluandosoftware.com/nota-3937-Best-Practices-en-sincronia-con-cada-organizacion.html>

Deming, E. (s.f.). Calidad, productividad y competitividad: la salida de la crisis.

Deporte, Ministerio de Educación, cultura y. (2012). La enseñanza de las ciencias en Europa: políticas nacionales, prácticas e investigación. Red Española de Información Sobre Educación, 176.

Desarrollo, C. (2010). CMMI ® para Desarrollo, Versión 1.3.

Digital, A. F. (2012). Alianza Futuro Digital Medellín Apuesta productiva y educativa en el sector software de Medellín (p. 82). Medellín. Recuperado el 16 de Junio de 2012, de: <http://issuu.com/viverodelsoftware/docs/libro-afdm-final>

- Eischen, K. (2005). Los servicios de tecnología de la información en India: protagonistas, lugares y prácticas. Recuperado de: <http://revistas.bancomext.gob.mx/rce/magazines/87/4/RCE.pdf>
- Fedesoft. (2012). Estudio de la caracterización de productos y servicios de la industria de software y servicios asociados (p. 78).
- Fernández, C. O et al (1995). Un enfoque actual sobre la calidad del software. Scielo. ACIMED v.3 n.3 Ciudad de La Habana. Recuperado de: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94351995000300005
- Fiti. (2013). Colombia entre los Países destacados en el uso de Prácticas de Calidad (SEI).
- Garbajosa, J (2006). Guide to the Software Engineering Body of Knowledge Swebok guía de proyectos. Madrid. Recuperado de: <http://www.calidaddelsoftware.com/documentos/II%20Semana%20CMMI/12-UPM-SWEBOK.pdf>
- Giandini, R y otros. (2009) A two –level formal semantics for the QVT language. Facultad de informática, Universidad Nacional de la Plata. Buenos Aires. Argentina
- Framework (2015) Me Rinde Venezuela - www.merinde.net
- Giraldo, F. D., & Cardona, S. (n.d.). Una propuesta de especificación de atributos de calidad para DSBC aplicada a un caso práctico de sistemas de información ambiental.

- Huff, M. P.-, & Cannon, R. (2009). The Personal Software Process (PSP) Body of Knowledge, Version 2.0 (p. 92).
- Humphrey, W. (1997), Introduction to the Personal Software Process, Addison Wesley Longman, Inc., Massachusetts.
- IEEE Computing & Control Engineering Journal. (1997), Vol. 8, No. 3, June pp.120-126.
- Ivanisevich, J., et al. (1997) Management Quality and Competitiveness, Second Edition, Irwin/MsGraw-Hill, New York.
- Kitchenham, B. et al (1997) “DESMET: A methodology for evaluating software engineering methods and tools” Computing & Control Engineering Journal.
- Kruchten, P. (2011). Experience teaching software project management in both industrial and academic settings. 2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEET), 199–208. doi:10.1109/CSEET.2011.5876087
- Machinery, A. for C., & Engineers, I. of E. and E. (1999). Código de Ética y Práctica Profesional 5.2.
- Mahnic, V. (2012). A Capstone Course on Agile Software Development Using Scrum. IEEE TRANSACTIONS ON EDUCATION, 55,N 1, 8.
- Marín, B et al. (2009) Identificación de defectos en modelos conceptuales utilizados en entornos MDA. Centro de investigación en métodos de producción de software, Universidad Politécnica de Valencia, España.

- Martínez T, et al (2009). Modelado de líneas de procesos mediante SPEM v2.0. Departamento de tecnología y Sistemas de información Escuela Superior de informática de Ciudad Real. Universidad de Castilla- La Mancha.
- Mendoza. L. E, & Pérez, M. E. S. B. (2005). Prototipo de modelo sistémico de calidad (MOSCA) del software. *Computación Y Sistemas*, 8 Núm 3, 196–217. Recuperado de: <http://www.scielo.org.mx/pdf/cys/v8n3/v8n3a5.pdf>
- Mora, B. (2009) SMML: Lenguaje para la representación de modelos de medición del software. Departamento de tecnología y Sistemas de información Escuela Superior de informática de Ciudad Real. Universidad de Castilla- La Mancha
- Moreno, J., & Bolaños, L. del C. (s.f.). Exploración de Modelos y Estándares de calidad para el producto software. *UIS Ingenierías Revista de La Facultad de Ingenierías Fisicomecánicas*.
- Moroni, N et al (2005). Estrategias para la enseñanza de la programación. *JEITICS 2005 - Primeras Jornadas de Educación en Informática y TICS en Argentina*.
- Ortega, M. et al (2000). "A Model for Software Product Quality with a Systemic Focus," in 4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2000 and The 6th International Conference on Information
- Pardo, C et al. (2009) Homogenización de marcos en ambientes de mejora de procesos multimarco. Facultad de Ingeniería Electrónica y Telecomunicaciones. Universidad del Cauca. Colombia
- Pérez, M., et al (2001) "Systemic Quality Model for System Development Process: Case Study"

Pfleeger.S.L (2002). "Ingeniería de Software. Teoría y Práctica". Prentice Hall.

Pierre N. et al (2001) "YOOPEEDOO (UPEDU): A Process for Teaching Software Process," cseet, pp.18, 14th Conference on Software Engineering Education and Training.

Pino, F. J. del C.-C., & García, F. C. _La M. (2006). Revisión sistemática de mejora de procesos en micro, pequeñas y medianas empresas. Revista Española de Innovación, Calidad E Ingeniería de Software, 19.

Pino, F et al. (2009) Aplicación del marco metodológico de Competisoft a través de investigación – acción y casos de estudio. Facultad de Ingeniería Electrónica y Telecomunicaciones. Universidad del Cauca. Colombia

Pressman. (2002)."Ingeniería del Software 5ª Edición". McGraw Hill.

Real Academia Española. (2014) Diccionario de la lengua española versión tricentenario (23 ed)

Rivera, J et al. (2009) Extending visual modeling languages with timed behavioral specifications. Universidad de Málaga. España.

Rosero, C. (2012) Estrategia metodológica para el seguimiento al desarrollo de competencias en estudiantes de la media técnica del proyecto Alianza Futuro Digital Medellín (AFDM).

Scrum manager. Glosario [en línea]. (Citada: 5 de noviembre de 2012) Recuperado de <http://www.scrummanager.net/oks/mod/glossary/view.php?id=158>

Schwaber, K., & Sutherland, J. (2013). La guía de Scrum. La guía definitiva de Scrum: Las reglas del juego (p. 19).

Sergio, C. del Q.-C., & Faria, J. P. of P. (2014). TSP Symposium 2013 Proceedings.

Seventh Americas Conference on Information Systems – AMCIS (2001) Boston, Massachusetts, Agosto 2001, pp. 1297-1304. Recuperado de:<http://www.lisi.usb.ve/publicaciones>

Society, I. C. (1998). IEEE 1061-1998 Standard for a Software Quality Metrics Methodology. Recuperado de <http://ieeexplore.ieee.org/servlet/opac?punumber=6061>

Society, I. C. (2014). Swebok Guide V3.0 (p. 335). Recuperado de <http://www.computer.org/portal/web/swebok>

Software Construction: A (2007) “ Software Construction and Maintenance ” Course and its Maintenance Corpus, 176–180.

Software Engineering Institute - Mellon University, C. (2014). Team Software Process. Recuperado de <http://www.sei.cmu.edu/tsp/tools/>

Software Engineering Institute. (2012) CMMI for development, version 1.3 Recuperado el 16 de Junio de 2012, de <http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm>

Sommerville. I (2002). “Ingeniería de Software” Sexta edición. Addison Wesley.

SQA. (2013a). Atributos de Calidad de Software. Recuperado de www.sqa.es

SQA. (2013b). SQA en la práctica. Recuperado de www.sqa.es

Systems, Analysis and Synthesis ISAS (2000), Orlando, Florida, Julio, 2000, pp. 395-401, <http://www.lisi.usb.ve/publicaciones>.

Tabares. L (2011) Personalización de RUP para proyectos académicos de desarrollo de software. Universidad Eafit.

Trazar, F. (s.f.). Buenas prácticas en desarrollo de software apuntes de una experiencia.

Tremblay, G., et al (2007). Introducing Students to Professional

Vanrell. J. A & Bertone, R. N. (s.f.). Un Modelo de Procesos de explotación de información. Universidad Tecnológica Nacional Recuperado de http://sedici.unlp.edu.ar/bitstream/handle/10915/19462/Documento_completo.pdf?sequence=1

Vargas, C (2009) Sistema para auditar el cumplimiento de CMMI-SW nivel 2 Capítulo 3 - Aseguramiento de la calidad del software. Recuperado de http://sedici.unlp.edu.ar/bitstream/handle/10915/3956/3_Aseguramiento_de_la_calidad_del_software.pdf?sequence=11

Velázquez, A. (2012). Mejores Prácticas para el Desarrollo de Software. SG Virtual. SG #37 (Agosto-Octubre 2012). Recuperado el 21 octubre de 2015, de <http://sg.com.mx/revista/mejores-pr%C3%A1cticas-para-el-desarrollo-software#.VjFbCNlvc1k>

Visconti. M. et al (1997). Experiencia con un modelo de madurez para el mejoramiento del proceso de aseguramiento de calidad del software. Universidad Técnica Federico Santa María. Valparaiso, Chile. Recuperado el 20 Julio de 2014, de <http://www.inf.utfsm.cl/~visconti/papers/papersqa1997.pdf>

Villalobos S.J.A & Casallas. R. (2006). Fundamentos de Programación Aprendizaje activo basado en casos.

Weitzenfeld. A. (2004). "Ingeniería de Software Orientado a Objetos con UML, Java e Internet". Thomson.

Zeiss, B. of G., & Vega, D. U. B. (s.f.). Applying the ISO 9126 Quality Model to Test Specifications Exemplified for TTCN-3 Test Specifications.

12 ANEXO: FORMULARIO - ENCUESTA A EMPRESARIOS SOBRE MEJORES PRÁCTICAS

ALIANZA FUTURO DIGITAL MEDELLÍN - AFDM



ALIANZA PARA LA “TRANSFORMACIÓN Y ARTICULACIÓN DE LA EDUCACIÓN TÉCNICA - MEDIA Y PROFESIONAL – Y TECNOLÓGICA CON EL CLUSTER DEL SOFTWARE Y AFINES”

Encuesta mejores prácticas en Calidad de software

Señor empresario

Con el fin de identificar las mejores prácticas que su organización usa para desarrollar software de calidad, le solicitamos contestar a las siguientes preguntas:

Componente organizacional				
1	Su organización se encuentra certificada en:			
		¿Cuánto hace que se encuentra certificado?		
		De 1 a 3 años <input type="checkbox"/>	De 3 a 5 años	Más de 5 años <input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/> CMMI nivel _____			
	<input checked="" type="checkbox"/> ISO 9001:2008			
	<input type="checkbox"/> ISO/IEC 9003Ingeniería de software			
<input checked="" type="checkbox"/> ISO/IEC 12207Information				

	Technology / Software Life Cycle Processes			
	ISO/IEC 15504 – SPICE (Software Process Improvement and Assurance Standards Capability Determination)			
	ISO/IEC 9126 Software engineering – Product quality			
	ISO/IEC 14598 Software product evaluation			
	<input type="checkbox"/> ISO 25000			
	<input type="checkbox"/> IT Mark			
	<input type="checkbox"/> TSP- Team Software Process			
	<input type="checkbox"/> PSP - Personal Software Process			
	<input type="checkbox"/> Otra: _____ _____			
2.	Enumere 3 ventajas que usted ha identificado para su organización a partir de su proceso de certificación			
3.	Enumere 3 ventajas que usted a identificado para su organización en el desarrollo de su producto/servicio software a partir de su proceso de certificación			

4.	Enumere cuáles son sus 3 mejores prácticas implementadas para que su personal desarrolle un buen producto software.
Componente Talento humano	
5.	Su organización realiza proceso de inducción al personal nuevo: SI <input checked="" type="checkbox"/> NO <input checked="" type="checkbox"/>
5.1	Es un proceso formal SI <input checked="" type="checkbox"/> NO <input checked="" type="checkbox"/>
5.2	¿Cuánto tiempo dura? De 1 a 3 semanas <input checked="" type="checkbox"/> De 3 a 6 semanas <input checked="" type="checkbox"/> Más de 6 semanas <input checked="" type="checkbox"/>
5.3	¿En qué consiste? ¿ Quién lo hace?
5.4	¿Su organización entrega un manual de funciones y procedimientos? SI <input type="checkbox"/> NO <input checked="" type="checkbox"/>
5.5	¿Ha contratado en su organización personas que hayan realizado su formación por ciclos propedéuticos acabados de salir del colegio (Media técnica)? SI <input checked="" type="checkbox"/> NO <input checked="" type="checkbox"/>
5.6	¿En la actualidad cuenta con personal que haya realizado su formación por ciclos propedéuticos (Media técnica)? SI <input checked="" type="checkbox"/> NO <input checked="" type="checkbox"/>
5.7	¿Cuáles son las razones por las cuales dicho personal ya no labora en su organización? En caso de no estar
	Falta idoneidad en competencias del ser
	Falta idoneidad en competencias del saber <input checked="" type="checkbox"/>
	Falta idoneidad en competencias del hacer <input checked="" type="checkbox"/>
	Falta idoneidad en competencias del convivir <input checked="" type="checkbox"/>
	Falta idoneidad en competencias técnicas
	Tuvo una propuesta mejor <input checked="" type="checkbox"/>
	Recorte de personal <input checked="" type="checkbox"/>
	Otra: ¿Cuál? <input checked="" type="checkbox"/>
5.8	Indique qué tipo de actividades les asigna a los jóvenes contratados que vienen del proceso de formación por ciclos propedéuticos
	Relacionadas con programación <input checked="" type="checkbox"/>
	Relacionadas con Helpdesk <input checked="" type="checkbox"/>

	Otra: ¿Cuáles? <input type="checkbox"/>																																																																								
	Promedio de tiempo que duran los jóvenes en la organización																																																																								
5.9	De acuerdo a las siguientes competencias que deben desarrollar los jóvenes en su proceso de formación, indique el grado de desarrollo evidenciado en el trabajo de los jóvenes, marcando de 1 (menor) a 5 (mayor). No solo aplica para los nuevos también para los ingenieros, personal en general																																																																								
	<table border="1"> <thead> <tr> <th>Competencia</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> </tr> </thead> <tbody> <tr> <td>Trabajo en equipo</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Orientación al logro</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Comunicación oral y escrita</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Investigación y creatividad</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Manejo de TIC</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Pensamiento analítico – sistémico</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Pensamiento algorítmico</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Comprensión organizacional</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Programar en un lenguaje</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Documentación de un software</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Construcción de modelado de datos</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Competencia	1	2	3	4	5	Trabajo en equipo						Orientación al logro						Comunicación oral y escrita						Investigación y creatividad						Manejo de TIC						Pensamiento analítico – sistémico						Pensamiento algorítmico						Comprensión organizacional						Programar en un lenguaje						Documentación de un software						Construcción de modelado de datos					
Competencia	1	2	3	4	5																																																																				
Trabajo en equipo																																																																									
Orientación al logro																																																																									
Comunicación oral y escrita																																																																									
Investigación y creatividad																																																																									
Manejo de TIC																																																																									
Pensamiento analítico – sistémico																																																																									
Pensamiento algorítmico																																																																									
Comprensión organizacional																																																																									
Programar en un lenguaje																																																																									
Documentación de un software																																																																									
Construcción de modelado de datos																																																																									
5.10	¿Qué cambios deberían darse dentro de la Alianza Futuro Digital para vincular a estos jóvenes?																																																																								

Componente Buenas prácticas					
6.	Señale cuáles de las siguientes prácticas usted implementa en su organización				
	6.1 Requisitos:				
	PRACTICAS	SI	NO	A VECES	SIEMPRE
	Identificar, definir y atender asuntos éticos, económicos, culturales, legales y ambientales relacionados a los proyectos de trabajo.				
	Participación activa de los clientes				
	Visualización de los requerimientos organizados por prioridad.				
	Comunicación asertiva con su equipo de trabajo, entrega de información de una sola fuente.				
	Visualización de la arquitectura, primero desde un nivel de abstracción alto para identificar una estrategia que permita la implementación de la solución.				
	Elaborar documentación entregable a través del ciclo de vida del producto de forma paralela a la creación de la solución				
	Ver más allá del modelado. Algunas veces los requerimientos que tienen una mayor prioridad son complejos, lo cual implica explorarlos antes de comenzar su desarrollo para reducir el riesgo general del desarrollo.				

6.2	¿Qué perfil de persona desarrolla esta actividad en su organización? Técnico <input type="checkbox"/> Tecnólogo <input type="checkbox"/> Profesional <input type="checkbox"/> Otro: <input type="checkbox"/> ¿Cuál? _____				
6.3 Análisis y diseño					
	PRACTICAS	SI	NO	A VECES	SIEMPRE
	El diseño general del sistema se construye conforme avanza el desarrollo del proyecto cambiando y evolucionando constantemente				
	Los modelos de diseño no se encuentran completos, los detalles se refinan durante el proceso de codificación.				
	Analizar detenidamente el ambiente de implementación.				
	Los diseñadores también deben codificar. Cuando el modelo es desarrollado por alguien más se entrega a otra persona para codificarlo hay un riesgo significativo de que no capte sus detalles adecuadamente. Separar las funciones del diseño y la codificación es riesgosa, es mejor contar con especialistas en el equipo que puedan diseñar y codificar.				
	Realizar análisis por riesgos				
	6.4 ¿Qué perfil de persona desarrolla esta actividad en su organización? Técnico <input type="checkbox"/> Tecnólogo <input type="checkbox"/> Profesional <input type="checkbox"/> Otro: <input type="checkbox"/> Todos <input type="checkbox"/> ¿Cuál? _____				
6.5 Desarrollo					
	PRACTICAS	SI	NO	A VECES	SIEMPRE

	Probar con código. Nunca debe asumirse que un diseño funciona, sino que debe probarse codificándolo para determinar si funciona				
	La retroalimentación es importante. Nunca debe olvidarse que es necesario buscar activamente retroalimentación sobre el trabajo que se realiza. Esto permite mejorar el sistema.				
	Utilizar herramientas de generación de código.				
	Documentar las partes complicadas del sistema. A través de la documentación generada debemos poder entender el funcionamiento del sistema, así como las razones que sustentan las decisiones de diseño.				
	Verificación y seguimiento de estándares				
	Adecuada planificación de la construcción y mantener mediciones acerca del desempeño y calidad de los programadores				
	Usar con propiedad los diferentes lenguajes que apoyan el proceso de construcción en los diferentes niveles de la arquitectura: lenguajes de presentación, lenguajes de programación, lenguajes de configuración, lenguajes de bases de datos				
6.6	¿Qué perfil de persona desarrolla esta actividad en su organización? Técnico <input type="checkbox"/> Tecnólogo <input type="checkbox"/> Profesional <input type="checkbox"/> Otro: <input type="checkbox"/> ¿Cuál? _____				
	6.7 Organizacionales				
	PRACTICAS	SI	NO	A VECES	SIEMPRE
	Revisar el trabajo de otros en forma objetiva, sincera y apropiadamente documentada.				

	Escuchar equitativamente las opiniones, preocupaciones y quejas de un colega.				
	Ayudar a sus colegas a que estén totalmente alertas a los actuales estándares incluyendo políticas y procedimientos de protección de contraseñas, archivos, información confidencial y las medidas de seguridad en general.				
	En situaciones fuera de sus propias áreas de competencia, solicitar las opiniones de otros profesionales que tengan competencia en esa área.				
	Mejorar su conocimiento de los avances en el análisis, especificación, diseño, desarrollo, mantenimiento, pruebas del software y documentos relacionados, junto con la administración del proceso de desarrollo				
	Mejorar su habilidad para producir documentación precisa, informativa y bien redactada.				
	Mejorar su comprensión del software de los documentos con que se trabaja y del medio ambiente donde serán usados				
	Calidad en el proceso dará calidad en el Producto.				
	Asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)				
	Uso de arquitectura basada en componentes				

	Control de cambios				
	Modelado visual del software				
	El aseguramiento de la calidad es parte del proceso de desarrollo y no la responsabilidad de un grupo independiente				
	Hay cultura organizacional, equipos auto-organizados, auto – control, auto –				
	Documentar lecciones aprendidas				
	Se realiza revisión entre pares				
6.8	¿Qué perfil de persona desarrolla esta actividad en su organización?				
	Técnico Tecnólogo Profesional Otro: ¿Cuál? _____				
7	Recomendaciones:				