



---

UNIVERSIDAD CATÓLICA LOS ÁNGELES  
CHIMBOTE

[www.uladech.edu.pe](http://www.uladech.edu.pe)

**RECTORADO**

# **METODOLOGÍA DE DESARROLLO DE SOFTWARE**

**VERSIÓN 001**

**CHIMBOTE – PERÚ**

**2017**

Versión: 001	Código: MDS	F. Implementación: 19-12-17	Pág.: 1 de 39
Elaborado por: División de Sistemas	Revisado por: Dirección de Calidad	Aprobado por Rectorado	

## COLABORADORES

En orden alfabético:

<b>Apellidos y Nombres</b>	<b>Cargo</b>
Enríquez Ruiz, José Luis	Coordinador de Tecnología Web
Farías Palacín, Elías	Desarrollador
Flores Flores, Eder	Desarrollador Líder
Honores Solano, Cesar	Coordinador de Transferencia Tecnológica Externa
Llanos Muñoz, Roger	Desarrollador Líder
López Cordero, William	Administrador de Base de Datos
Medina Luna, Victoria Oscar	Coordinador de Transferencia Tecnológica Interna
Olivos Colchado, Carlos	Desarrollador
Torres Quito, Carlos	Desarrollador
Velásquez Soto, Gilmer	Jefe de Sistemas
Zúñiga Ángeles, Albert	Desarrollador

Versión: 001	Código: MDS	F. Implementación: 19-12-17	Pág.: 2 de 39
Elaborado por: División de Sistemas	Revisado por: Dirección de Calidad	Aprobado por Rectorado	

## CONTENIDO

<b>INTRODUCCIÓN</b>	5
<b>Capítulo I - Conceptos Generales</b>	6
1.1. Términos y Definiciones	6
1.2. Alcance	8
1.3. Objetivos de la Metodología	8
1.4. Principios	9
1.5. Componentes de la Metodología	9
1.6. Roles	11
1.6.1. Coordinador	11
1.6.2. Desarrollador Líder	12
1.6.3. Equipo de Desarrollo	12
1.6.4. Administrador de Base de Datos (DBA)	13
1.6.5. Comité de Calidad	14
1.7. Documentos de Control	15
1.7.1. Acta de Requerimiento	15
1.7.2. Formato de Resultados de Análisis de Requerimiento de Usuario (F3)	15
1.7.3. Formato de Conformidad de Cambios en Base de Datos (F4)	16
1.7.4. Cronograma de Actividades	16
1.7.5. Formato de Pruebas de Interfaces y Contenido (F1)	17
1.7.6. Formato de Validación de Requerimientos (F2)	17
1.7.7. Acta de Puesta en Producción	17
1.7.8. Acta de Cierre	18
<b>Capítulo II - El Ciclo de Vida de la Metodología</b>	18
2.1. Análisis	19
2.1.1. Descripción de actividades	20
2.1.2. Documentos de Control	21
2.2. Diseño	21
2.2.1. Descripción de actividades	22
2.2.2. Documentos de Control	22
2.3. Codificación	22
2.3.1. Descripción de Actividades	23
2.3.2. Documentos de Control	23
2.4. Pruebas	23
2.4.1. Descripción de Actividades	24
2.4.2. Documentos de Control	25
2.5. Implementación	25
2.5.1. Descripción de Actividades	26

Versión: 001	Código: MDS	F. Implementación: 19-12-17	Pág.: 3 de 39
Elaborado por: División de Sistemas	Revisado por: Dirección de Calidad	Aprobado por Rectorado	

2.5.2.	Documentos de Control.....	26
ANEXOS	.....	27

Versión: 001	Código: MDS	F. Implementación: 19-12-17	Pág.: 4 de 39
Elaborado por: División de Sistemas	Revisado por: Dirección de Calidad		Aprobado por Rectorado

## INTRODUCCIÓN

En el campo del desarrollo de software, existen dos grupos de metodologías, las denominadas tradicionales (formales) y las ágiles.

Las primeras son un tanto rígidas, exigen una documentación exhaustiva y se centran en cumplir con el plan del proyecto definido totalmente en la fase inicial del desarrollo del mismo; mientras que la segunda enfatiza el esfuerzo en la capacidad de respuesta a los cambios, las habilidades del equipo y mantener una buena relación con el usuario.

Ambas propuestas tienen sus propias ventajas y desventajas; de cualquier manera, las metodologías de desarrollo nos dicen el ¿Qué hacer? más no el ¿Cómo hacer?, esto significa que la metodología que elijamos, debe ser adaptada al contexto del proyecto, teniendo en cuenta los recursos técnicos y humanos; tiempo de desarrollo y tipo de sistema.

A lo largo de todo este tiempo; los proyectos de software que han “nacido” en nuestra casa de estudios, no fueron concebidos bajo la guía de alguna metodología reconocida en el medio; la improvisación ha sido una característica muy frecuente a la hora de abordar los proyectos; sin embargo estas carencias eran compensadas por la experiencia de algunos desarrolladores que fueron los que forjaron las bases para ordenar el trabajo; para ello se utilizaron algunos procedimientos, guías de estandarización y convenciones de codificación que ayudaron a establecer el orden en el desarrollo de los productos.

En este documento se plantea una propuesta metodológica para el desarrollo de software; orientada a promover el desarrollo con resultados rápidos y que no involucre un alto grado de complejidad; que sea de fácil adaptación y que permita implementar los controles necesarios para desarrollar productos de software estándares y de calidad.

En el primer capítulo se incluye la terminología, definiciones y conceptos necesarios para familiarizarse con el propósito del presente documento. En el segundo capítulo; se describen las fases y las actividades que conforman el ciclo de desarrollo del software; así mismo se presenta los documentos y formatos de control que se deberán utilizar durante todo el desarrollo del proyecto. Y por último se incluyen los modelos de formatos en la sección anexos.

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 5 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

## CAPÍTULO 1

### CONCEPTOS GENERALES

#### 1.1. Términos y Definiciones

- **Área de desarrollo.-** Se refiere a una unidad operativa dentro de la DISI en la cual se desarrollan productos de software. Por ejemplo: Coordinación de Transferencia Tecnológica Interna, Coordinación de Transferencia Tecnológica Externa y Coordinación de Tecnología Web son áreas de desarrollo.
- **Cliente.-** Es la persona o empresa receptora de un bien, servicio, producto o idea, a cambio de dinero u otro artículo de valor
- **Coordinador.-** Es el responsable de gestionar el proyecto. Para mayor información revise la sección Roles.
- **Desarrollador.-** Es un especialista en informática que es capaz de concebir y elaborar sistemas informáticos; así como de implementarlos y ponerlos a punto, utilizando el lenguaje de programación definido. Para mayor información revise la sección Roles.
- **Desarrollador Líder.-** Tiene como función realizar el seguimiento y el control del cronograma de actividades. Para mayor información revise la sección Roles.
- **Entregables.-** Son los productos intermedios que generan las fases. Los entregables permiten evaluar la marcha del proyecto mediante comprobaciones de su adecuación o no a los requisitos funcionales y de condiciones de realización previamente establecidos.
- **Método.-** Un método se compone de diversos aspectos que nos permitirán conseguir una meta o lograr un objetivo. Se define más claramente como un conjunto de herramientas, las cuales, utilizadas mediante las técnicas correctas, permiten la ejecución de procesos que nos llevarán a cumplir los objetivos que buscamos
- **Metodología.-** Hace referencia al conjunto de procedimientos racionales utilizados para alcanzar el objetivo o la gama de objetivos que rige una investigación científica, una exposición doctrinal o tareas que requieran habilidades, conocimientos o cuidados específicos. Con frecuencia puede definirse la metodología como el estudio o elección de un método pertinente o adecuadamente aplicable a determinado objeto.
- **Metodología de Desarrollo de Software.-** Es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. En un proyecto de desarrollo de software la metodología ayuda a definir: *Quién* debe hacer *Qué* *Cuándo* y *Cómo* debe hacerlo.  
  
La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende actividades a seguir

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 6 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado.

- **Módulo Informático.-** Es una porción de software que realiza tareas específicas dentro del Erp University. Por ejemplo: módulo de matrícula.
- **Proyecto.-** Según la definición que nos proporciona PMI en su guía PMBOOK, un proyecto se podría definir como “un servicio temporal que se lleva a cabo para crear un producto, servicio o resultado único”

Podemos decir entonces que un proyecto tiene un inicio y un fin, este fin se tiene que alcanzar dentro de un tiempo fijado.

- **Software.-** IEEE Std. 610 define el software como “programas, procedimientos, documentación y datos asociados, relacionados con la operación de un sistema informático”

El software se puede definir como el conjunto de tres componentes:

Programas (instrucciones): este componente proporciona la funcionalidad deseada y el rendimiento cuando se ejecute.

Datos: este componente incluye los datos necesarios para manejar y probar los programas y las estructuras requeridas para mantener y manipular estos datos.

Documentos: este componente describe la operación y uso del programa.

#### **Componentes del Software:**

Es importante contar con una definición exhaustiva del software ya que de otra manera se podrían olvidar algunos componentes. Una percepción común es que el software sólo consiste en programas. Sin embargo, los programas no son los únicos componentes del software.

#### **Programas**

Los programas son conjuntos de instrucciones que proporcionan la funcionalidad deseada cuando son ejecutadas por el ordenador. Están escritos usando lenguajes específicos que los ordenadores pueden leer y ejecutar, tales como lenguaje ensamblador, Basic, FORTRAN, COBOL, C... Los programas también pueden ser generados usando generadores de programas.

#### **Datos**

Los programas proporcionan la funcionalidad requerida manipulando datos. Usan datos para ejercer el control apropiado en lo que hacen. El mantenimiento y las pruebas de los programas también necesitan datos. El diseño del programa asume la disponibilidad de las estructuras de datos tales como bases de datos y archivos que contienen datos.

#### **Documentos**

Además de los programas y los datos, los usuarios necesitan también una explicación de cómo usar el programa.

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 7 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

Documentos como manuales de usuario y de operación son necesarios para permitir a los usuarios operar con el sistema.

Los documentos también son requeridos por las personas encargadas de mantener el software para entender el interior del software y modificarlo, en el caso en que sea necesario.

- **Stakeholders.-** Es una palabra del inglés que, en el ámbito empresarial, significa ‘interesado’ o ‘parte interesada’, y que se refiere a todas aquellas personas u organizaciones afectadas por las actividades y las decisiones de una empresa.

En toda organización, además de sus propietarios, participan diversos actores claves y grupos sociales que están constituidos por las personas o entes que, de una manera y otra, tienen interés en el desempeño de una empresa porque están relacionadas directa o indirectamente, con ella.

- **Usuario.-** Es un usuario quien tiene un conjunto de permisos y de recursos (o dispositivos) a los cuales se tiene acceso. Es decir, un usuario puede ser tanto una persona como una máquina, un programa, etc.

Para efecto del presente documento se utilizará “Usuario” para referirse a la persona con la que se establece comunicación para tratar temas relacionados a los requerimientos.

## 1.2. Alcance

La metodología deberá ser aplicada en todas las unidades de desarrollo de software de la Universidad; esto incluye:

- Transferencia Tecnológica Interna
- Transferencia Tecnológica Externa
- Tecnología Web
- Otros equipos que por su naturaleza no están incluidos dentro de las tres anteriores.

Asimismo la metodología podrá ser transferida al Equipo de TI de las instituciones que han adquirido el Erp University según los acuerdos estipulados en los contratos.

Deberá ser de conocimiento y alcance a las unidades de auditorías o áreas a fines dentro de la Universidad.

## 1.3. Objetivos de la Metodología

La metodología busca cumplir con los siguientes objetivos:

- Asegurar la uniformidad y calidad tanto del desarrollo como del sistema en sí.
- Satisfacer las necesidades de los usuarios del sistema.
- Conseguir un mayor nivel de rendimiento y eficiencia del personal asignado al desarrollo.
- Ajustarse a los plazos y costes previstos en la planificación.

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 8 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	



- Facilitar el mantenimiento posterior de los sistemas.
- Definir actividades a llevarse a cabo en un Proyecto de Sistema de Información.
- Unificar criterios en la organización para el desarrollo del Sistema de Información.
- Proporcionar puntos de control y revisión.
- Permitir construir un sistema documentado y que sea fácil de mantener.
- Ayudar a identificar, lo antes posible, cualquier cambio que sea necesario realizar dentro del proceso de desarrollo.

#### 1.4. Principios

La metodología recoge algunos de los principios ágiles propuestos por el “Agile Manifiesto”.

El “Manifiesto Ágil” incluye cuatro postulados y una serie de principios asociados. Sus postulados son:

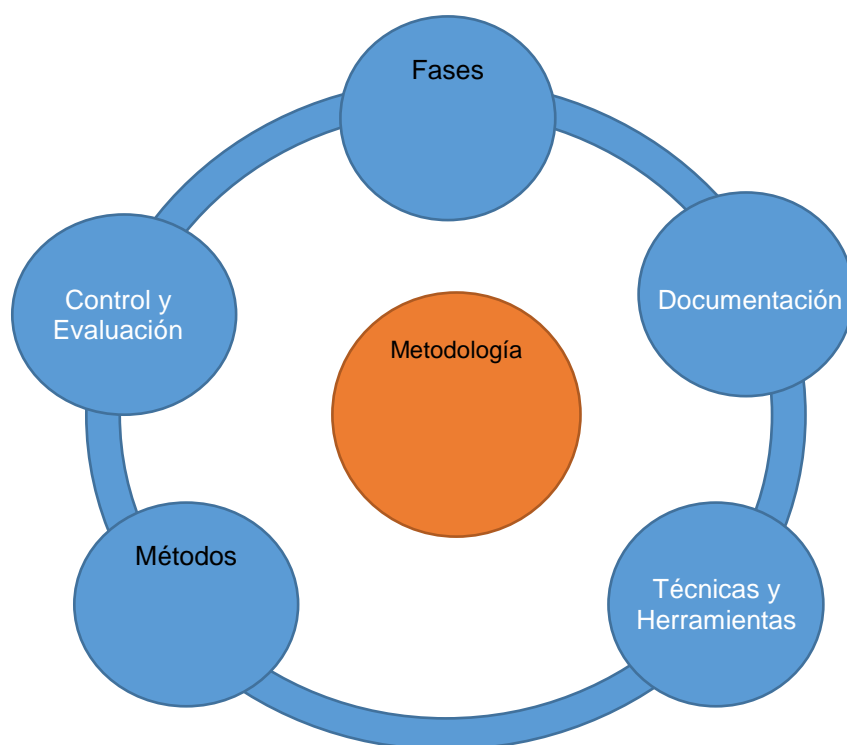
- **Valorar al individuo y a las interacciones del equipo de desarrollo por encima del proceso y las herramientas.** Tres premisas sustentan este principio: a) los integrantes del equipo son el factor principal de éxito de un proyecto; b) es más importante construir el equipo de trabajo que construir el entorno; y c) es mejor crear el equipo y que éste configure el entorno en base a sus propias necesidades.
- **Valorar el desarrollo de software que funcione por sobre una documentación exhaustiva.** El principio se basa en la premisa que los documentos no pueden sustituir ni ofrecer el valor agregado que se logra con la comunicación directa entre las personas a través de la interacción con los prototipos. Se debe reducir al mínimo indispensable el uso de documentación que genera trabajo y que no aporta un valor directo al producto.
- **Valorar la colaboración con el cliente por sobre la negociación contractual.** En el desarrollo ágil el cliente participa y colabora con el equipo de trabajo como si fuese un integrante más. Los requerimientos de usuario en sí no aporta valor al producto, es sólo un formalismo que establece líneas de responsabilidad entre las partes.
- **Valorar la respuesta al cambio por sobre el seguimiento de un plan.** La evolución rápida y continua deben ser factores inherentes al proceso de desarrollo. Se debe valorar la capacidad de respuesta ante los cambios por sobre la capacidad de seguimiento y aseguramiento de planes pre-establecidos.

#### 1.5. Componentes de la Metodología

La metodología se define como la disciplina que indicará qué métodos y técnicas hay que usar en cada fase del ciclo de vida del desarrollo del proyecto.

Los elementos que componen la metodología son:

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 9 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	



**Figura 1: Elementos básicos de la Metodología**

**Fases.-** Son etapas del proceso de desarrollo de software. En la metodología se identificarán las diferentes actividades que se realizarán en cada fase.

Una fase es un conjunto de actividades relacionadas con un objetivo en el desarrollo del proyecto. Se construye agrupando tareas (actividades elementales) que pueden compartir un tramo determinado del tiempo de vida de un proyecto. La agrupación temporal de tareas impone requisitos temporales correspondientes a la asignación de recursos (humanos, financieros o materiales).

La metodología contempla las fases de: Análisis, Diseño, Desarrollo, Pruebas e Implementación; cada una de estas etapas se describen más adelante.

**Métodos.-** Es necesario identificar el modo en el que se realizará el proceso de desarrollo del producto software. Se tendrá que descomponer los procesos en actividades más pequeñas, en estas actividades se definen los valores que recibirá cada fase así como los que generará y la técnica que se tendrá que usar.

**Técnicas y Herramientas.-** Indican como se deberá de resolver cada actividad y que herramientas podríamos usar. Existen diferentes tipos de técnicas, algunas de ellas son:

- **De recopilación de datos:** entrevistas, formularios, etc.
- **Técnicas gráficas:** Diagramas, organigramas, matrices, etc.
- **Técnicas de modelado:** Desarrollo estructurado y orientado a objetos.

**Documentación.-** Es necesario especificar qué documentación se va generar durante cada etapa del proceso; estos documentos deben realizarse de manera completa y usando todos los valores de entrada y salida que se van generando, esto servirá para recoger los resultados y tomar decisiones de las diferentes situaciones planteadas. Por ejemplo; Actas de Reuniones, Formatos de Pruebas, etc.

**Control y Evaluación.-** Las actividades de control y evaluación se deben de realizar a lo largo de todas las fases para identificar errores y corregirlos a tiempo. En resumen, consiste en realizar el seguimiento del avance de acuerdo al cronograma de trabajo; puede ser necesario tomar decisiones como el replanteamiento de la planificación de las tareas asignadas para lograr los objetivos propuestos.

## 1.6. Roles

La metodología propone cinco roles principales; cada uno de estos cumplen un papel muy particular dentro del proceso de desarrollo de software. Estos son los siguientes:

- Coordinador
- Desarrollador Líder
- Equipo de desarrollo
- Administrador de base de datos
- Comité de calidad

### 1.6.1.Coordinador

El Coordinador; es la persona que toma las decisiones, y es el que realmente debe conocer el negocio del cliente y su visión del producto. El Coordinador es el responsable de gestionar la Lista de Requerimientos del Usuario, el cual incluye:

- Expresar claramente los elementos de la Lista de Requerimientos del Usuario. Es el encargado de interpretar lo que el usuario necesita en un lenguaje claro y preciso.
- Ordenar los elementos en la Lista de Requerimientos del Usuario para alcanzar los objetivos y misiones de la mejor manera posible.
- Optimizar el valor del trabajo desempeñado por el Equipo de Desarrollo, planificando el tiempo de ejecución.
- Asegurar que la Lista de Requerimientos es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación.

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 11 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

### 1.6.2.Desarrollador Líder

El Desarrollador Líder es un facilitador por naturaleza; tiene las siguientes responsabilidades:

- Es el que analiza los requerimientos del usuario y propone la solución. Define las actividades que se deberán hacer para desarrollar el producto; así mismo es el responsable de identificar el alcance del producto y los requerimientos necesarios para su implementación.
- Diseña el modelo de datos y socializa con el Administrador de Base de Datos antes de su implementación. El objetivo, es asegurar que el modelo propuesto no contravenga las convenciones establecidas por el DBA y las buenas prácticas en el diseño de base de datos.
- Se reúne con su equipo de desarrolladores para dar a conocer el avance del proyecto y las actividades que se vienen desarrollando con el objetivo de generar el compromiso de parte de los desarrolladores.
- Fomenta el compañerismo dentro de su equipo de trabajo y en su entorno para mejorar la empatía entre todos los desarrolladores.
- Realiza el seguimiento del avance a su equipo de trabajo para asegurarse que el entregable será terminado dentro del tiempo establecido en el cronograma de trabajo.
- Revisa los entregables terminados por parte de su equipo de trabajo, para ello utiliza el formato F1. Su función es asegurarse que los entregables cumplan con las especificaciones, y que se cumplan los criterios de calidad expuestos en el formato F1.
- Concertar con el Coordinador para establecer estrategias que permitan mejorar la relación y el bienestar de todo el equipo; creando espacios de relajamiento con el objetivo de incrementar la productividad de cada integrante.
- Velar por el cumplimiento de los valores y principios ágiles, las reglas y guiar la colaboración en el equipo y con el usuario. Esto significa que debe asegurar que exista una lista de requisitos priorizadas y que esté lista antes de iniciar cada iteración. Así mismo, debe facilitar las reuniones del equipo para revisar los avances en base a demostraciones de manera que se pueda cumplir con los objetivos.

### 1.6.3.Equipo de Desarrollo

El equipo de Desarrollo, son los responsables de desarrollar el producto siguiendo las indicaciones del Desarrollador Líder y basándose en el Cronograma de Actividades.

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 12 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

El equipo de desarrollo cuenta con todas las habilidades necesarias para la creación de un producto; tiene las siguientes responsabilidades:

- En base a las actividades asignadas y las indicaciones respectivas de parte del Desarrollador Líder, llevan a cabo las acciones para desarrollar el producto entregable. Para ello ponen en práctica todos sus conocimientos y habilidades al servicio del proyecto.
- Crean o modifican objetos de la base de datos (tablas, procedimientos, funciones, etc.) siguiendo las indicaciones de parte del Desarrollador Líder.
- Realizan pruebas unitarias del producto para asegurarse que cada componente funcional (función, procedimiento, rutina, etc.) cumpla con el objetivo y devuelva los resultados esperados.
- Realizan pruebas de funcionalidad de todo el producto entregable para asegurarse que no hay errores de codificación. En esta parte también se revisa que el nuevo entregable no tenga efecto negativo en el resto de módulos.
- Luego de asegurarse que el producto entregable es funcional y no presenta errores; se procede a subir los archivos al repositorio (versionador) y posteriormente deberá desplegar en la instancia de pruebas.
- Los Desarrolladores son capaces de innovar y proponer estrategias para resolver los casos que se presenten en el quehacer diario.
- Los equipos deben ser pequeños como para permanecer ágil y lo suficientemente grande como para completar una cantidad de trabajo significativa dentro del tiempo establecido. Se recomienda que los equipos tengan como mínimo 2 y como máximo 9 integrantes.

Los Equipos de Desarrollo más pequeños podrían encontrar limitaciones en cuanto a las habilidades necesarias para entregar a tiempo un producto. Tener más miembros de lo recomendado requiere demasiada coordinación. Los Equipos de Desarrollo grandes generan demasiada complejidad para que pueda gestionarse mediante procesos empíricos. El Coordinador y el Desarrollador Líder no cuentan en el cálculo del tamaño del equipo a menos que también estén contribuyendo a trabajar en el Cronograma de Actividades.

#### **1.6.4.Administrador de Base de Datos (DBA)**

El Administrador de base de datos tiene un papel importante dentro del proceso de desarrollo de software; su función es asegurar que la Base de datos cumpla

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 13 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

con los requisitos técnicos requeridos por el producto entregable y brindar todas las facilidades al área de desarrollo para acceder a las base de datos.

Entre otras responsabilidades, el DBA tiene las siguientes:

- Asegurar que las instancias de base de datos estén disponibles y accesibles según las demandas por parte de las áreas de desarrollo. Estas instancias incluyen las de pruebas, capacitaciones y las instancias de producción que están a su alcance.
- Definir las políticas de acceso a las base de datos por parte del equipo de desarrollo de software de acuerdo a las políticas establecidas dentro la DISI y de la propia universidad.
- Facilitar los medios para realizar actualizaciones en los objetos de la base de datos según corresponda. Esto debe incluir: la frecuencia, el responsable y la forma como se enviarán los scripts.
- Mantener actualizado el diccionario de datos; para ello deberá establecer conjuntamente con el área de desarrollo, los mecanismos para mantenerse informado de las actualizaciones realizadas por parte del área de desarrollo.
- Dar conformidad al modelo de datos propuesto por el área de desarrollo mediante el *Formato de Conformidad de Cambios en Base de Datos*.

#### 1.6.5. Comité de Calidad

El Comité de Calidad está compuesto por cualquier integrante del equipo de desarrollo; así mismo en ciertas ocasiones pueden participar personas externas, incluido el mismo usuario. Se debe tomar en cuenta en no incluir en el Comité de Calidad a quienes están involucrados directamente con el entregable; por cuestiones de transparencia y objetividad, el desarrollador y su Desarrollador Líder no pueden conformar el Comité de Calidad.

Es importante señalar que los integrantes del Comité de Calidad están familiarizados con el proceso de negocio y tienen experiencia en desarrollo de software; estas cualidades permitirán que las decisiones que tomen serán fundamentadas y no dará lugar a objeciones.

El Comité será designado por el Coordinador que corresponde de acuerdo al Módulo Informático en cuestión.

Las responsabilidades del Comité de Calidad son las siguientes:

- Validar el cumplimiento de los requerimientos del usuario; basándose en el *Acta de Requerimiento*; el Comité de Calidad deberá asegurarse que en el entregable se hayan considerado todos los requerimientos del usuario. En esta parte, el Comité de Calidad se deberá apoyar en el *Formato de Validación de Requerimientos de Usuario (F2)*

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 14 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

- Verificar la operatividad de los módulos, teniendo en cuenta las condiciones del cliente y estableciendo observaciones según corresponda.
- Firmar el *Formato de Validación de Requerimientos de Usuario*.

## 1.7. Documentos de Control

Los documentos de control evidencian el trabajo que se está haciendo en cada una de las etapas del proceso de desarrollo; permiten realizar el seguimiento de las actividades y establecen hitos en cada fase.

De acuerdo a la naturaleza de los requerimientos del usuario, se pueden utilizar algunos o todos los documentos de control.

Los documentos de control son los siguientes:

- Acta de Requerimiento
- Formato de Resultados de Análisis de Requerimiento de Usuario (F3)
- Formato de Conformidad de Cambios en Base de Datos (F4)
- Cronograma de Actividades
- Formato de Pruebas de Interfaces y Contenido (F1)
- Formato de Validación de Requerimientos (F2)
- Acta de Puesta en Producción
- Acta de Cierre

### 1.7.1. Acta de Requerimiento

Este documento debe contener la lista de requerimientos acordado con el usuario en la reunión para el levantamiento de los requerimientos. La descripción de cada requerimiento debe ser clara y precisa, sin emplear palabras técnicas para que cualquier usuario pueda comprender lo que se va a hacer.

El Coordinador es el responsable de llenar este documento en cooperación con el Usuario. No deben considerarse fechas de entrega para evitar falsas expectativas.

El Acta de Requerimiento deberá ser firmada por el usuario y por el Coordinador.

El formato del Acta de Requerimiento se incluye en el **Anexo 01**.

### 1.7.2. Formato de Resultados de Análisis de Requerimiento de Usuario (F3)

Producto del análisis realizado, se plantea la solución al problema; así mismo los requerimientos necesarios para su desarrollo; esto incluye: software, hardware, recurso humano especializado, asesoría, capacitaciones, entre otros.

El formato de este documento se incluye en el **Anexo 02**.

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 15 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

#### 1.7.3. Formato de Conformidad de Cambios en Base de Datos (F4)

Este documento es utilizado por el Administrador de Base de Datos para aprobar los modelos de datos creados por las áreas de desarrollo. El objetivo de esta verificación es asegurar que el modelo de datos propuesto, cumpla con las especificaciones y los estándares establecidos por el DBA.

El formato puede encontrarlo en el **Anexo 03**.

#### 1.7.4. Cronograma de Actividades

El Cronograma de Actividades contiene el detalle de las actividades definidas de acuerdo al requerimiento del usuario; es decir en base al Acta de Requerimiento; el responsable de elaborar este documento es el Desarrollador Líder con el apoyo del Coordinador.

En este documento deben considerarse los siguientes datos:

- **Proyecto.-** Indicar el proyecto al cual corresponde la actividad.
- **Módulo.-** Indicar el módulo al cual pertenece la actividad.
- **Nro. Requerimiento:** Indica el número de requerimiento que da origen al cronograma de actividades.
- **Código de la Actividad.-** Debe ser único en todo el documento; se recomienda utilizar este dato en los diálogos o consultas dentro del equipo para facilitar la identificación de cada actividad.
- **Estado.-** Se deben manejar los siguientes estados: Sin empezar, En Progreso, Terminado, Revisión, Cerrado.

Estado	Descripción
Sin empezar	Indica que la actividad se encuentra en espera para ser asignado a algún desarrollador
En progreso	Indica que el desarrollador está trabajando en el desarrollo de la actividad.
Terminado	Indica que el desarrollo de la actividad ha concluido y está listo para ser revisado.
Revisión	Indica que el resultado de la actividad está siendo revisada para asegurar que se han cumplido con las especificaciones del requerimiento del usuario.
Cerrado	Indica que el resultado de la actividad ha pasado todos los criterios de validación y de calidad y está listo para instalarse en el ambiente de producción.
Cancelado	Indica que la actividad ha sido cancelada y que ya no se realizará.



- **Descripción de la actividad.-** Debe ser clara y precisa; evitar utilizar palabras técnicas para que personas externas puedan entender lo que se va a hacer.
- **Desarrollador.-** El nombre del desarrollador asignado a la actividad.
- **Fecha de Inicio.-** La fecha cuando se inicia el desarrollo de la actividad.
- **Fecha de Término.-** La fecha cuando se concluye la actividad.
- **Etapas.-** Para indicar en qué etapa se encuentra la actividad el cual debe ser actualizado a lo largo del desarrollo.
- **Nivel de dificultad.-** Especificar el nivel de dificultad de la actividad de acuerdo a: Simple, Media, Compleja.  
El nivel de dificultad es determinado en base a los criterios que establezca el Desarrollador Líder basándose en su experiencia y la habilidad del desarrollador. El nivel de dificultad permitirá establecer el tiempo de desarrollo de la actividad.
- **Prioridad.-** Indicar el nivel de prioridad: Baja, Media, Alta.

El Formato del Cronograma de Actividades se incluye en el **Anexo 04**.

#### 1.7.5. Formato de Pruebas de Interfaces y Contenido (F1)

Este formato se utiliza para validar el cumplimiento de algunos criterios de estandarización relacionados con: Diseño, Usabilidad, Seguridad, Base de Datos y Código Fuente. Debe aplicarse luego que el desarrollador ha culminado con el entregable y antes de pasar a la instancia de producción.

Este formato se incluye en el **Anexo 05**.

#### 1.7.6. Formato de Validación de Requerimientos (F2)

Tomando como referencia el Acta de Requerimiento del usuario, se deberá utilizar el Formato de Validación de Requerimientos para asegurar que se están considerando todos los requerimientos del usuario; de tal manera que no haya objeciones de parte del usuario durante la presentación del entregable.

El modelo de este formato se incluye en el **Anexo 06**.

#### 1.7.7. Acta de Puesta en Producción

Permite dejar constancia que el producto entregable se ha instalado en el ambiente de producción y que está listo para ser utilizado por el usuario. Se debe especificar los responsables, datos del entregable, información del ambiente o entorno donde se ha instalado; así mismo se debe indicar los componentes que se están instalando (Base de datos, código fuente, etc.)

El modelo de este formato se incluye en el **Anexo 07**.

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 17 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

### 1.7.8. Acta de Cierre

Este documento se deberá elaborar únicamente cuando el Módulo Informático se ha concluido y debe hacerse la entrega de la documentación respectiva.

Usualmente, este documento es utilizado en los proyectos externos, donde el mantenimiento del mismo será de responsabilidad del Cliente.

En este documento se debe detallar las funcionalidades y las actividades que se realizaron para desarrollar el Módulo Informático. De la misma manera indicar los documentos complementarios que se le hace entrega al usuario.

El modelo de este formato se incluye en el **Anexo 08**.

## CAPÍTULO 2

### EL CICLO DE VIDA DE LA METODOLOGÍA

El ciclo de vida es el conjunto de fases por las que un producto (entregable) deberá pasar; se inicia con el Análisis y termina en la fase de Implementación.

Entre las funciones que debe tener el Ciclo de Vida se destacan las siguientes:

- Determinar el orden de las fases del proceso de software.
- Establecer los criterios de transición para pasar de una fase a otra.
- Definir las entradas y salidas de cada fase.
- Describir los estados por los que pasa el entregable.
- Describir las actividades a realizar para transformar el producto.
- Definir un esquema que sirve como base para planificar, organizar y coordinar.

El ciclo de vida se compone de fases sucesivas compuestas por actividades que se pueden planificar. Según el modelo de ciclo de vida, la sucesión de fases puede ampliarse con bucles de retroalimentación, de manera que lo que conceptualmente se considera una misma fase se pueda ejecutar más de una vez a lo largo de un proyecto, recibiendo en cada pasada de ejecución, aportaciones a los resultados intermedios que se van produciendo (retroalimentación).



**Figura 2: Ciclo de vida del desarrollo**

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 18 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

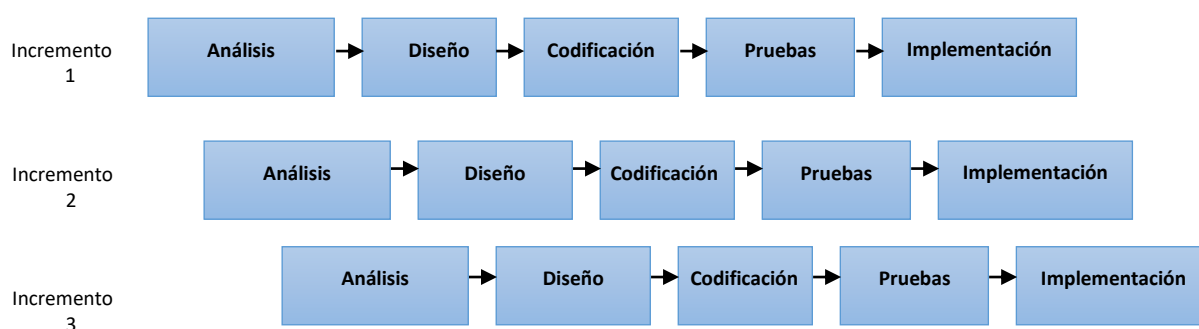
El modelo de ciclo de vida de software es una vista de las actividades que se deben realizar durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociados entre estas etapas.

El modelo de ciclo de vida del software:

- Describe las fases principales de desarrollo de software.
- Define las fases primarias esperadas de ser ejecutadas durante esas fases.
- Ayuda a administrar el progreso del desarrollo.
- Provee un espacio de trabajo para la definición de un proceso detallado de desarrollo de software.

En cada una de las etapas del ciclo de vida, se establecen una serie de actividades que deben ser realizadas por cada participante en el proyecto, desde el Coordinador hasta el Desarrollador.

El ciclo de vida de la metodología se basa en el modelo incremental; la filosofía de este modelo consiste en incrementar las funcionalidades del software en varias iteraciones. Este modelo aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software.



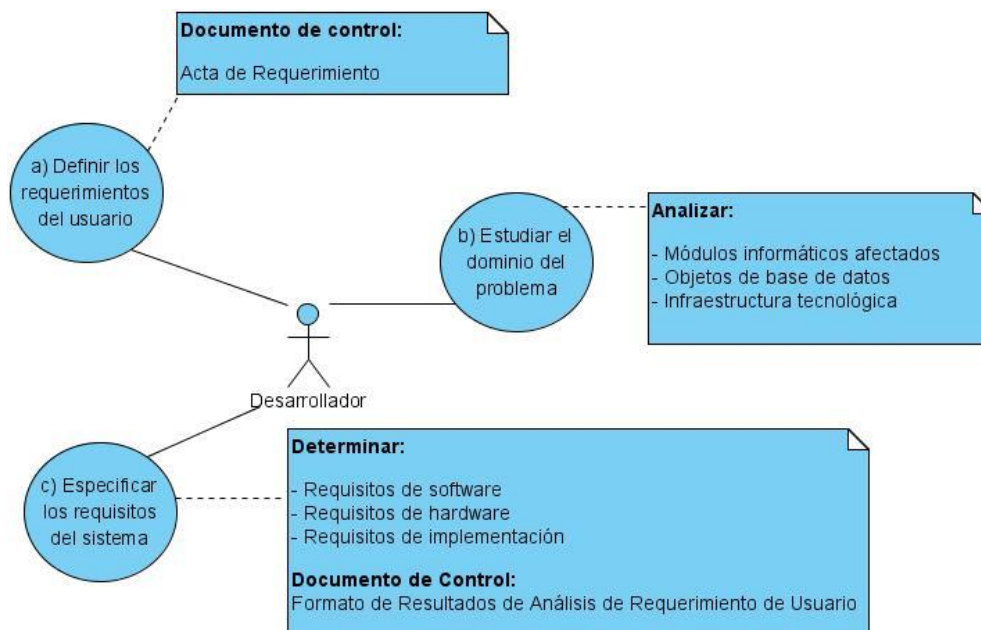
**Figura 3: Modelo del Proceso de desarrollo de Software**

## 2.1. Análisis

Esta etapa es la más importante del ciclo de desarrollo; dependiendo del trabajo que se realice aquí, se podrá comprender la naturaleza del problema correctamente; aquí se determina que es lo que realmente se necesita hacer. Es una etapa crítica, por ello se requiere la participación de personas con experiencia; porque de no realizarse un buen análisis puede traer consecuencias negativas para el proyecto, principalmente en el cumplimiento a tiempo.

En el siguiente diagrama se puede apreciar las actividades que se desarrollan en la etapa de Análisis:

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 19 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	



**Figura 4: Actividades de la Etapa de Análisis**

### 2.1.1.Descripción de actividades

En esta parte se describen las actividades que se realizan durante el análisis:

**a) Definir los requerimientos del usuario;** para ello es necesario identificar las personas y áreas de la organización afectadas con el problema; también es importante elegir correctamente las técnicas o herramientas que se utilizarán para facilitar la recopilación de la información; se pueden utilizar entrevistas, cuestionarios; optar por observar el funcionamiento normal del entorno e inclusive participar activamente en él (por ejemplo desempeñando temporalmente el trabajo de los usuarios). También podría utilizarse un formato de solicitud de requerimientos estándar que será llenado por el mismo usuario que luego será revisado por los analistas.

Al término de esta actividad, el analista deberá presentar el *Acta de Requerimiento* debidamente llenado y firmado por el usuario.

**b) Estudiar el dominio del problema;** el cual consiste en identificar relaciones con otros módulos o componentes del producto y evaluar las implicancias que conllevaría realizar modificaciones en objetos de la base de datos o en el código fuente. Así mismo, se evalúan los recursos que se necesitan para su implementación, tales como infraestructura tecnológica (Servidores, equipos, etc.). Puede que el analista requiera el apoyo de las áreas de redes o soporte técnico.

De la misma manera se debe asegurar que el requerimiento no vaya a infringir alguna norma jurídica o alguna política interna; para ello deberá apoyarse de otras áreas de la organización tales como asesoría legal.

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 20 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

c) **Especificar los requisitos del sistema;** consiste en describir en forma detallada los requisitos de software y hardware para poder implementar el producto. En esta parte pueden considerarse los siguientes aspectos:

- **Requerimientos Funcionales.-** Describen las interacciones entre el producto y su entorno, usuarios u otros sistemas (módulos).

Por ejemplo:

- El sistema deberá validar el número de documento de identidad mediante un dígito de verificación.
- Luego de guardarse los datos del nuevo docente, el sistema deberá enviar un correo electrónico con los datos de su cuenta de acceso al correo personal.

- **Requerimientos no Funcionales.-** Describen aspectos visibles del producto por el usuario, y que no se relacionan directamente con comportamiento funcional del producto. Por ejemplo: Tiempos de respuesta, memoria RAM necesaria y usabilidad; así mismo el lenguaje de programación, tipo de aplicación (desktop, web, móvil), entre otros.

- **Requerimientos de Implementación.-** Corresponden a las necesidades del cliente que restringen la implementación, como la plataforma tecnológica, de hardware, redes, etc.

Al final de esta actividad, se deberá presentar el Formato de Resultado de Análisis de Requerimiento de Usuario (F3).

### 2.1.2.Documentos de Control

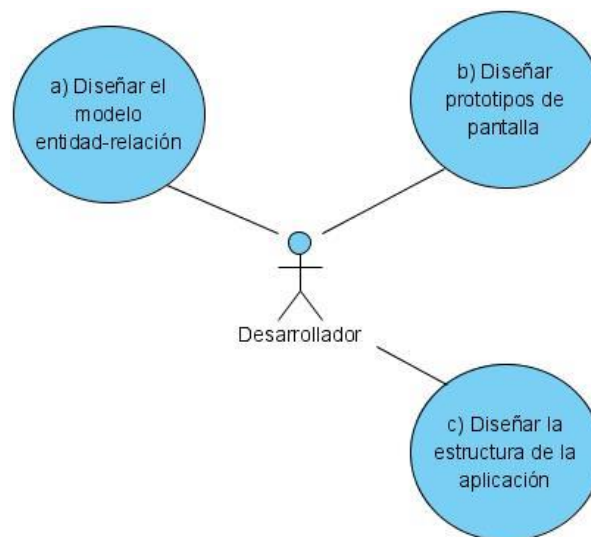
- Acta de Requerimiento.
- Formato de Resultados de Análisis de Requerimiento de Usuario (F3).

## 2.2.Diseño

En esta fase, el desarrollador utiliza la información obtenida en el Análisis y elabora el diseño lógico del producto.

El diseño se enfoca en cuatro atributos; (1) la estructura de los datos, (2) la arquitectura del software, (3) el detalle procedimental y (4) la caracterización de la interfaz.

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 21 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	



**Figura 5: Actividades de la Etapa de Diseño**

### 2.2.1.Descripción de actividades

- a) **Diseñar el modelo entidad-relación;** para ello se utiliza la herramienta de modelado y se diseña cada una de las entidades y atributos principales que van a necesitarse para almacenar los datos. Probablemente, para algunos requerimientos no se necesite modificar la Base de Datos; en ese sentido esta parte puede obviarse.
- b) **Diseñar prototipos de pantalla;** los prototipos ayudan a normalizar las funcionalidades del sistema y permiten construir interfaces uniformes.
- c) **Diseñar la estructura de la aplicación;** se debe considerar la jerarquía de módulos establecidos para el producto que se va a desarrollar. En esta parte se debe identificar que archivos son los que se van a crear y de qué tipo; así mismo la distribución de los mismos dentro de cada carpeta del proyecto.

### 2.2.2.Documentos de Control

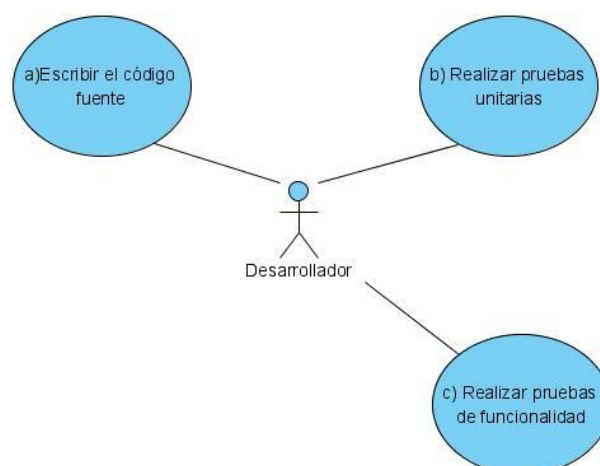
- Prototipos de Pantallas.
- Formato de Conformidad de Cambios en Base de Datos (F4)

### 2.3.Codificación

Una vez que se cuenta con los documentos de control de la fase de Análisis y del Diseño; se inicia la fase de Codificación. Evidentemente para codificar se necesita conocer la sintaxis del lenguaje de programación que se vaya a emplear.

En esta parte, el desarrollador deberá seguir los lineamientos impuestos en el Diseño y tomando en consideración siempre los requisitos funcionales y no funcionales.

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 22 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	



**Figura 6: Actividades de la Etapa de Codificación**

### 2.3.1.Descripción de Actividades

- a) **Escribir el código fuente;** en esta parte, el desarrollador deberá asegurarse que durante la escritura del código fuente está siguiendo las normas y convenciones de codificación.
- b) **Realizar pruebas unitarias;** consiste en probar la funcionalidad de una parte del código fuente, tales como rutinas, funciones, etc. con el objetivo de asegurar que los resultados devueltos sean los correctos.
- c) **Realizar pruebas de funcionalidad;** una vez que se tenga una versión terminada del producto, el desarrollador debe realizar pruebas para asegurarse que las entradas definidas producen los resultados esperados y que todos los componentes del producto funcionan correctamente

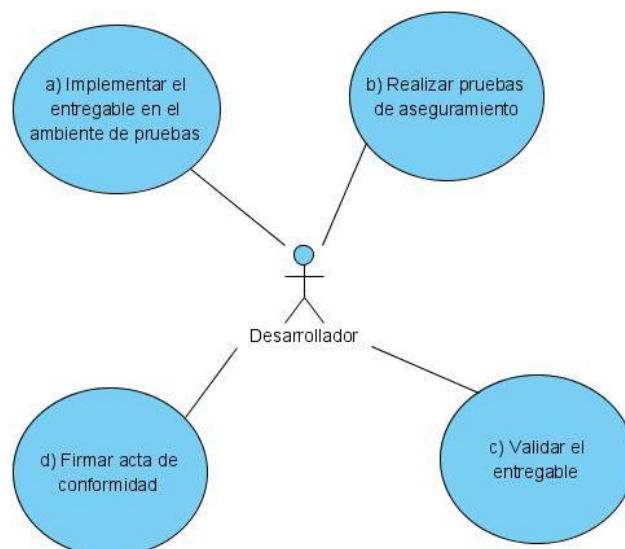
### 2.3.2.Documentos de Control

- Formato de Pruebas de Interfaces y Contenido (F1)

## 2.4.Pruebas

Luego que el producto se ha terminado de codificar; debe ser instalado en el ambiente de pruebas. En esta parte se busca comprobar que el producto funciona correctamente y que cumple con los requerimientos del usuario. Las pruebas finales del producto son realizadas por el mismo usuario con la guía del desarrollador.

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 23 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	



**Figura 7: Actividades de la Etapa de Pruebas**

#### **2.4.1.Descripción de Actividades**

- a) Implementar el producto en el ambiente de pruebas;** el desarrollador deberá asegurarse que todos los componentes del producto se instalen correctamente para evitar que durante las pruebas se presenten errores o inconvenientes, esto incluye: objetos de base de datos, archivos de código fuente, datos iniciales en las tablas y otros recursos necesarios. Aquí también debe asegurarse que se instale la infraestructura necesaria.  
Es importante que la instancia de prueba tenga la misma configuración de los servicios (servidores) que el servidor de producción para detectar cualquier error en este punto y no cuando pase a producción.
- b) Realizar pruebas de aseguramiento;** el desarrollador deberá realizar las pruebas de funcionalidad en el ambiente de pruebas, para asegurarse que el producto se ha instalado correctamente.
- c) Validar el entregable;** el desarrollador deberá exponer el nuevo entregable frente al Comité de Calidad con el objetivo de obtener la conformidad de que se han cumplido todos los requerimientos del usuario especificados en el Acta de Requerimiento. En este evento, es posible que se presenten observaciones de parte del Comité de Calidad que posteriormente deberán ser corregidas antes de pasar a firmar el Formato de Validación de Requerimientos (F2)
- d) Firmar acta de conformidad;** siempre y cuando no existan observaciones sobre el entregable; se debe proceder a firmar el Formato de Validación de Requerimientos (F2) por parte del Comité de Calidad.

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 24 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	



### 2.4.2.Documentos de Control

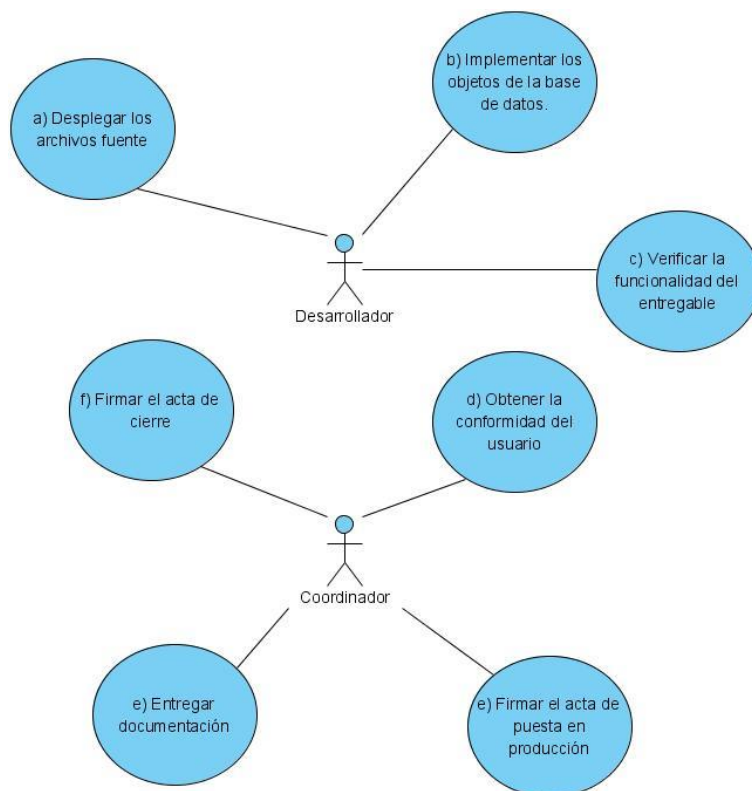
- Formato de Validación de Requerimientos (F2)

## 2.5.Implementación

Tomando como punto de partida el modelo de la fase anterior, se procede a programar o implementar el producto entregable. El propósito de esta etapa es instalar el software y los requisitos necesarios para que el entregable pueda correr.

En esta fase, se consideran las siguientes actividades:

- Desplegar los archivos fuente
- Implementar los objetos de la base de datos.
- Verificar la funcionalidad del entregable
- Obtener la conformidad del usuario
- Firmar el acta de puesta en producción
- Entregar documentación
- Firmar el acta de cierre



**Figura 8: Actividades de la Etapa de Implementación**

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 25 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

### 2.5.1.Descripción de Actividades

- a) **Despliegue de los archivos;** consiste en subir los archivos de código fuente en el repositorio intermedio; para ello se puede utilizar herramientas como TortoiseSVN o el Plug-in del IDE de desarrollo.
- b) **Actualización de la Base de Datos;** el desarrollador deberá enviar los scripts al responsable de la Base de Datos para su ejecución. En esta parte deben considerarse los datos iniciales de las tablas. Probablemente, el responsable en actualizar la Base de Datos este delegado a otra persona, en cuyo caso deberá realizarse según los procedimientos establecidos para tal fin.
- c) **Verificación del entregable en el ambiente de producción;** el desarrollador debe asegurarse que todo se ha subido correctamente haciendo las pruebas respectivas en el ambiente de producción.

### 2.5.2.Documentos de Control

- Acta de Puesta en Producción
- Acta de Cierre

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 26 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

# ANEXOS

## ANEXO 01: ACTA DE REQUERIMIENTO

 ULADECH UNIVERSIDAD ALGORA LOS ANGELES CUNDINAMARCA	ACTA DE REQUERIMIENTOS DE USUARIO
--	-----------------------------------

N° UNI-REQ-0001-2017

Cliente : UNIVERSIDAD \_\_\_\_\_  
 Módulo : (NOMBRE DE MÓDULO)  
 Responsable :

Fecha	Hora Inicio	Hora Término

N°	AGENDA
1	

N°	ACUERDOS
1	
2	
3	

N°	PARTICIPANTE	CARGO
1		
2		
3		
4		
5		

\_\_\_\_\_  
Usuario

\_\_\_\_\_  
Coordinador de Transferencia  
Tecnológica

## ANEXO 02: FORMATO DE RESULTADOS DE ANÁLISIS DE REQUERIMIENTO DE USUARIO (F3)



### FORMATO DE RESULTADOS DE ANÁLISIS DE REQUERIMIENTO DE USUARIO

I. DATOS DEL PROYECTO					
Código del Proyecto:	<Identificación del proyecto>	Nombre del Dominio:	<Nombre del dominio>	Nombre del Módulo:	<Nombre del Módulo Informático>
Responsable:	<Apellidos y nombres del analista>			Fecha de Ejecución:	<Fecha inicial> <Fecha término>

II. DATOS DEL REQUERIMIENTO			
Código del Requerimiento	<Identificación del Requerimiento>	Fecha de Solicitud:	<Fecha del requerimiento>
Descripción del Requerimiento	<Descripción del requerimiento que se está analizando, considera las condiciones o restricciones establecidas por el usuario.>		
Usuario:	<Apellidos y nombres del usuario responsable del requerimiento>		

III. ARQUITECTURA DE SOLUCIÓN	
Descripción de la Solución	<Describe la propuesta de solución, utilice un lenguaje natural y respondiendo a las siguientes preguntas: ¿Qué se va hacer?, ¿Cómo se hará?, ¿Con qué componentes del producto se relaciona?, ¿Qué recursos externos se necesitan?, ¿Cuáles son las condiciones o restricciones para su desarrollo o implementación?>
Arquitectura de la Solución	< Describe los componentes de la propuesta de solución, de ser necesario, incluya algún diagrama para reforzar su explicación>
Requisitos de Software	<Detalle el software necesario para poder desarrollar o implementar la propuesta de solución>



## FORMATO DE RESULTADOS DE ANÁLISIS DE REQUERIMIENTO DE USUARIO

Requisitos de Hardware	<Detalle el hardware necesario para poder desarrollar o implementar la propuesta de solución>
Otros requisitos	<Describe otros requisitos tales como: personal especializado, ambientes, asesorías, etc.>
Componentes	<Describe la relación de la propuesta de solución con otros componentes del producto>

IV. MODELO DE DATOS	
Diagrama del modelo de datos	<Incluya el diagrama del modelo de datos para la propuesta de solución. En el modelo deberá incluir todas las tablas que se utilizarán independientemente si ya están creadas o no>
Definición de nuevas tablas	<Detalle las nuevas tablas que se crearán para la propuesta de solución, describa que datos almacenará cada una de las tablas. Considere los campos principales de cada tabla. Asegúrese de utilizar los nombres tal cual se crearan en la base de datos.>
Definición de scripts	<Describe los procedimientos almacenados, funciones, triggers que se utilizan en la propuesta de solución>
Otras consideraciones	<Describe otros detalles sobre el modelo de datos que deben considerarse durante el desarrollo o en la implementación del entregable>

## ANEXO 03: FORMATO DE CONFORMIDAD DE CAMBIOS EN BASE DE DATOS (F4)



### FORMATO DE CONFORMIDAD DE CAMBIOS EN BASE DE DATOS

I. INFORMACIÓN GENERAL					
Proyecto:	<a href="#">&lt;Identificación del Proyecto&gt;</a>	Dominio:	<a href="#">&lt;Identificación del Dominio&gt;</a>	Módulo:	<a href="#">&lt;Identificación del Módulo&gt;</a>
Responsable:	<a href="#">&lt;Responsable de la presentación o exposición del entregable&gt;</a>				

II. DATOS DEL CAMBIO			
Tipo de Cambio	Cambio en base de datos	Código de requerimiento:	<a href="#">&lt;Código identificador del requerimiento asociado&gt;</a>
Descripción del cambio	<a href="#">&lt;Descripción del cambio, se detallará los motivos por los cuales se realiza el cambio en base de datos&gt;</a>		
Usuario:	<a href="#">&lt;Apellidos y nombres del usuario responsable del requerimiento&gt;</a>		

III. LISTA DE CHEQUEO DE CAMBIOS EN BASE DE DATOS					
	Cambio correcto				Observaciones
	SI	NO	NO APLICA		
<b>DENOMINACION DE TABLAS</b>					
1) ¿La tabla cuenta con prefijo asociado al módulo y esta separado por un guion bajo?					<a href="#">&lt;Puede incluir fechas de cumplimiento para levantar la observación&gt;</a>
2) ¿El nombre de la tabla está escrita en minúscula?					
3) ¿El nombre de la tabla indica la información que será almacenada en ella?					
4) ¿La longitud del nombre de la tabla no excede los 50 caracteres?					
<b>ATRIBUTOS DE TABLAS</b>					
1) ¿Los nombres de los campos están en minúscula?					
2) ¿Los nombres de los campos no exceden los 30 caracteres?					
3) ¿Se identifica la llave primaria?					
4) ¿Los campos que hacen referencia a otra tabla se encuentran relacionados mediante llave foránea?					
5) ¿La tabla cuenta con campos de auditoría (usuario y fecha de cambios en registro)?					

Página 1 de 3



### FORMATO DE CONFORMIDAD DE CAMBIOS EN BASE DE DATOS

6) El tipo de dato asociado al campo, así como la longitud de campo, ¿Corresponde al tipo de dato que se registrará?				
7) Los campos en la tabla ¿Cuentan con una descripción que ayude a orientar al desarrollador sobre el uso del mismo?				
8) Las colaciones establecidas tanto en tabla y campos, ¿Corresponden a la colación establecida en la base de datos?				
9) Los atributos especiales (autoincrement, Not null, unsigned,default,zero fill,etc) aplicados a los campos de una tabla ¿Tienen sustento para ser asignados?				
10) Los índices de las tablas ¿Se crean de acuerdo a necesidad del software?				
<b>SCRIPTS (PROCEDIMIENTOS ALMACENADOS, FUNCIONES, TRIGGERS, VISTAS, EVENTOS)</b>				
1) La programación almacenada en base de datos ¿Cuenta con el prefijo que lo identifique como tal?				
2) El nombre de la programación almacenada en base de datos ¿Describe de manera breve su funcionalidad?				
3) La programación almacenada en base de datos ¿Cuenta con un comentario dentro de la programación que permita entender su funcionalidad?				
4) El definidor de la programación almacenada en base de datos, se encuentra con el usuario 'root@localhost'?				
5) Los atributos de las variables y las tablas generadas en la programación almacenada en base de datos, se encuentran correctamente definidos?				

Página 2 de 3

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 31 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	



### FORMATO DE CONFORMIDAD DE CAMBIOS EN BASE DE DATOS

6) En caso de procedimientos almacenados, el tiempo de ejecución ¿No excede los 10 segundos?				
7) Las colaciones la programación almacenada en base de datos ¿Corresponden a la colación establecida en la base de datos?				
8) En el caso de los eventos, la duración del mismo ¿Es proporcional a la frecuencia en la que se ejecuta?				
<b>3.1. Resultados de la prueba</b>				
Defectos y desviaciones				Veredicto
<Observaciones>				<input type="checkbox"/> Paso
				<input type="checkbox"/> Falló
<b>3.2. Firma</b>				
<Apellidos y nombres del verificador>				



## ANEXO 04: CRONOGRAMA DE ACTIVIDADES

## CRONOGRAMA DE ACTIVIDADES

PROYECTO		MODULO		NRO -REQ	
----------	--	--------	--	----------	--

[illegible]

## ANEXO 05: FORMATO DE PRUEBAS DE INTERFACES Y CONTENIDOS (F1)



### FORMATO DE PRUEBAS DE INTERFACES Y CONTENIDOS

I. DATOS DEL PROYECTO			
Proyecto:	<Identificación del Proyecto>	Dominio:	<Identificación del Dominio>
Módulo:	<Identificación del Módulo>		
Responsable:	<Responsable de la presentación o exposición del entregable>		

II. INFORMACIÓN DEL CASO DE PRUEBA			
Tipo de Prueba:	Pruebas de Interfaces y Contenidos	Código Requerimiento:	<Código identificador del requerimiento asociado>
Descripción de la prueba:	<Descripción del objetivo de la prueba, detalle aquí lo que se va a realizar>		
Versión de Ejecución	<Versión o número de iteración de ejecución de la prueba>	Fecha:	<Fecha diligenciado por el tester>

#### 2.1. Pre-requisitos de la prueba

<Lista de los prerequisites a tener en cuenta antes de ejecutar la prueba>

#### 2.2. Lista de chequeo de la prueba

	Prueba satisfactoria			Observaciones
	SI	NO	NO APLICA	
<b>DISEÑO</b>				
1) ¿Los títulos o subtítulos de la página siguen las especificaciones del Manual del Desarrollador?				<Puede incluir fechas de cumplimiento para levantar la observación>
2) ¿La ubicación de los controles (cajas de texto, etiquetas, listas, botones, etc.) siguen las especificaciones del Manual del Desarrollador?				
3) ¿Se utilizan los íconos e imágenes de acuerdo a las especificaciones del Manual del Desarrollador?				
4) ¿Las etiquetas de los formularios o páginas web siguen las especificaciones predefinidas?				
5) ¿Existe consistencia en los enlaces (no existen enlaces rotos)?				



## FORMATO DE PRUEBAS DE INTERFACES Y CONTENIDOS

6) ¿Los estilos que se utilizan están definidos dentro del tema del diseño pre-establecido (hojas de estilo)?				
7) ¿Se utiliza una correcta ortografía en cada formulario o página web?				
<b>USABILIDAD</b>				
8) ¿Se incluyen leyendas para ayudar a entender el propósito de los iconos de acciones?				
9) ¿Se utiliza mensajes claros y descriptivos?				
10) ¿Se presenta al usuario información que solo necesita?				
11) ¿Se proporciona información visual de donde está el usuario (rutas de navegación)?				
12) ¿Se utiliza JavaScript para validar formularios durante su llenado y antes de enviarlos al servidor?				
<b>SEGURIDAD</b>				
13) ¿Existe ofuscación de parámetros y valores en las URLs de los enlaces?				
14) ¿Se evita visualizar las rutas absolutas de los directorios que evidencien la organización de los archivos en el servidor?				
15) ¿Se ha realizado saneamiento en el ingreso de datos para evitar ataques por inyección de código?				
16) ¿Se envían por POST los datos de los formularios para registro?				
17) ¿Se evita el registro o la consulta de datos mediante suplantación de identidad por doble inicio de sesión?				
<b>BASE DE DATOS</b>				
18) ¿Los nombres de los objetos de base de datos (tablas, campos, etc.) siguen las especificaciones del Manual del Desarrollador?				
19) ¿Se colocan comentarios sobre la funcionalidad de los scripts (procedimientos, funciones, triggers, eventos, etc.)?				

Página 2 de 3

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 35 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	



## FORMATO DE PRUEBAS DE INTERFACES Y CONTENIDOS

20) ¿Se utilizan transacciones en procedimientos que realizan tareas complejas?				
<b>CODIGO FUENTE</b>				
21) ¿Se utilizan comentarios en el código fuente para facilitar el entendimiento del mismo?				
22) ¿Los nombres de objetos (clases, funciones, etc.) siguen las especificaciones del Manual del Desarrollador?				
23) ¿El nombre de parámetros y variables siguen las especificaciones del Manual del Desarrollador?				
24) ¿Los archivos se han nombrado siguiendo las especificaciones del Manual del Desarrollador?				
25) ¿La ubicación de los archivos dentro de los directorios siguen las especificaciones del Manual del Desarrollador?				
<b>2.3. Resultados de la prueba</b>				
<b>Defectos y desviaciones</b>				<b>Veredicto</b>
<p style="color: blue; font-style: italic;">&lt;Lista de defectos o desviaciones encontrados por el tester&gt;</p>				<input type="checkbox"/> <b>Paso</b>  <input type="checkbox"/> <b>Falló</b>
<b>2.4. Equipo de Calidad</b>				
<b>Apellidos y Nombres</b>				<b>Firma</b>

## ANEXO 06: FORMATO DE VALIDACIÓN DE REQUERIMIENTOS



### FORMATO DE VALIDACION DE REQUERIMIENTOS

I. INFORMACIÓN GENERAL			
Proyecto:	<Identificación del Proyecto>	Dominio:	<Identificación del Dominio>
		Módulo:	<Identificación del Módulo>
Responsable:	<Responsable de la presentación o exposición del entregable>		

II. INFORMACION DEL CASO DE PRUEBA			
Tipo de Prueba:	Validación de Requerimientos	Código Requerimiento:	<Código identificador del requerimiento asociado>
Descripción de la prueba:	<Coloque en que consiste la prueba >		
Versión de Ejecución	1	Fecha:	<Fecha de la exposición>

#### 2.1. Pre-requisitos de la prueba

<Lista de los prerequisites a tener en cuenta antes de ejecutar la prueba>

Por ejemplo:

- El expositor presentará el módulo tres casos de prueba para el requerimiento 2 debido para facilitar la verificación del requerimiento 3 y 4.

#### 2.2. Lista de chequeo de la prueba

Lista de Requerimientos	Prueba satisfactoria		Observaciones
	SI	NO	
1)			<Puede incluir fechas de cumplimiento para levantar la observación>
2)			
3)			
4)			

#### 2.3. Resultados de la prueba

Observaciones y/o Recomendaciones	Veredicto
	<input type="checkbox"/> Paso <input type="checkbox"/> Falló

#### 2.4. Equipo de Calidad

Apellidos y Nombres	Firma

## ANEXO 07: ACTA DE PUESTA EN PRODUCCIÓN



### ACTA DE PUESTA EN PRODUCCIÓN

CONTRATO DE DESARROLLO DEL SISTEMA DE INFORMACIÓN PARA LA GESTIÓN ACADÉMICO - ADMINISTRATIVA "ERP UNIVERSITY" ENTRE LA UNIVERSIDAD CATÓLICA LOS ÁNGELES DE CHIMBOTE Y LA UNIVERSIDAD \_\_\_\_\_.

#### MÓDULO DE \_\_\_\_\_

##### I. CONTROL DE REVISIÓN DE PASE A PRODUCCIÓN

###### A. INVOLUCRADOS

Responsable del Módulo	
Fecha de Puesta en Producción	
Dueño de Proceso	

##### II. DATOS DEL SISTEMA DE INFORMACIÓN

###### A. GENERALES

Alcance del Módulo	
Objetivo del Módulo	
Proceso Involucrado	

###### B. INFORMACIÓN TÉCNICA DEL MÓDULO

APLICACIÓN EN ENTORNO WEB	
Dominio	
Base de Datos	
Lenguaje de Programación	
Directorio de Clases	
Directorio de Interfaces y Reportes	

##### III. ENTREGABLES A PRODUCCIÓN

- Base de Datos: Tablas, procedimientos, funciones, eventos y disparadores (triggers).
- Archivos fuente.

##### IV. OBSERVACIONES

El módulo es puesto en producción con el objetivo de iniciar el proceso de capacitación y asesoramiento en la configuración a los dueños del proceso, donde éstos a través de una serie de reuniones operarán la aplicación con la guía del capacitador designado por el ERP University, logrando una transferencia tecnológica satisfactoria.

Siendo el los representantes de la UNIVERSIDAD \_\_\_\_\_, y la Universidad Católica Los Ángeles de Chimbote - ULADECH aceptan la presente acta dando conformidad a la puesta en producción del módulo de REMUNERACIONES en cumplimiento parcial al contrato.

Representante de la Universidad  
POR LA RECEPCIÓN

Jefe de la División de Sistemas - ULADECH Católica  
POR LA ENTREGA

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 38 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	

## ANEXO 08: ACTA DE CIERRE



CONTRATO DE DESARROLLO DEL SISTEMA DE INFORMACIÓN PARA LA GESTIÓN ACADÉMICO - ADMINISTRATIVA "ERP UNIVERSITY" ENTRE LA UNIVERSIDAD CATÓLICA LOS ÁNGELES DE CHIMBOTE Y LA UNIVERSIDAD \_\_\_\_\_.

### ACTA DE CIERRE DEL MÓDULO DE \_\_\_\_\_

El día \_\_\_\_\_, el y el "(Representante del Cliente) \_\_\_\_\_" por parte de la UNIVERSIDAD \_\_\_\_\_, y la Auditora de Calidad del Sistema de Información y Comunicación \_\_\_\_\_, y el Jefe de la División de Sistemas \_\_\_\_\_, por parte de la UNIVERSIDAD CATÓLICA LOS ÁNGELES DE CHIMBOTE - ULADECH Católica, firman el acta de cierre del Módulo de \_\_\_\_\_ correspondiente al dominio de \_\_\_\_\_, en cumplimiento al documento Contrato N° \_\_\_\_\_ suscrito entre ambas instituciones con las características técnicas y especificaciones como se detalla a continuación:

#### 1. DETALLES DE LA ENTREGA DEL MÓDULO

##### 1.1. ALCANCE DEL MÓDULO

##### 1.2. ETAPAS DE DESARROLLO

Para cumplir con las etapas de desarrollo del módulo, se realizaron una serie de reuniones por videoconferencia (Skype) evidenciadas a través de actas donde participaron los dueños de proceso y personal del área de \_\_\_\_\_ de la UNIVERSIDAD \_\_\_\_\_ y el equipo de desarrollo del ERP University.

ETAPAS	CUMPLIMIENTO
REQUERIMIENTO DE USUARIO (Requisitos de Diseño)	100%
CODIFICACIÓN	100%
PUESTA EN PRODUCCIÓN	100%
IMPLEMENTACIÓN ASISTIDA (Capacitación, Configuración y Operatividad)	100%

#### 2. ENTREGABLES

- Manual de Usuario.
- Modelo (Diagrama ER) y Diccionario de Datos.

#### 3. SERVICIOS POST PRODUCCIÓN

La ULADECH Católica asume el compromiso de atender los futuros servicios de: consultoría, actualización, y soporte que el cliente requiera. Estos servicios serán atendidos dependiendo de la complejidad, disponibilidad de recursos y de factores regulatorios. Estos costos serán asumidos por el cliente, previo acuerdo de las partes.

Siendo el los representantes de la UNIVERSIDAD \_\_\_\_\_ y la Universidad Católica Los Ángeles de Chimbote - ULADECH Católica aceptan la presente acta dando conformidad al cierre del módulo de \_\_\_\_\_ en cumplimiento parcial al contrato.

Representante de la Institución  
POR LA RECEPCIÓN

Jefe de la División de Sistemas - ULADECH Católica  
POR LA ENTREGA

Dueño de Proceso  
POR LA RECEPCIÓN

Auditor de Calidad del SIC - ULADECH Católica  
POR LA ENTREGA

Versión: 002	Código: PRLD	F. Implementación: 19-12-17	Pág.: 39 de 3
Elaborado por: Comisión de Talento Docente	Revisado por: Dirección de Calidad	Aprobado con Resolución N° 1270-2017-CU-ULADECH CATÓLICA	