

Universidad de Alcalá



Escuela Politécnica Superior

**Master Universitario en Dirección de Proyectos de
Informáticos**

Trabajo Fin de Master

**Estudio sobre el análisis de estimación de un
proyecto IOT**

Autor: D. JAVIER ARTAL LORENTA

Julio 2019

Universidad de Alcalá

Escuela Politécnica Superior

**Master Universitario en Dirección de Proyectos de
Informáticos**

Trabajo Fin de Master

**ESTUDIO SOBRE EL ANÁLISIS DE
ESTIMACIÓN DE UN PROYECTO IOT**

Autor : D. JAVIER ARTAL LORENTA

Director Master : Dr. D. Roberto Barchino Plata

Tribunal evaluador :

Presidente del Tribunal :

Vocal 1º:

Vocal 2º:

Calificación : _____

Alcalá de Henares a, 12 de Julio del 2019



Universidad de Alcalá

Escuela politécnica superior

**MÁSTER UNIVERSITARIO EN DIRECCIÓN DE PROYECTOS
INFORMÁTICOS**

Trabajo Fin de Máster

**Estudio sobre el análisis de estimación de un
proyecto IOT**

Javier Artal Lorenta

Julio 2019

AGRADECIMIENTOS

A Cristina por todo el apoyo que siempre me ha ofrecido y por estar siempre a mi lado en todo momento.

A mis padres por seguir dándome oportunidades en la vida.

ÍNDICE

1	INTRODUCCIÓN.....	5
2	DESCRIPCIÓN DEL PROYECTO.....	7
3	OBJETIVOS.....	8
4	ESTADO DEL ARTE.....	9
5	IDENTIFICACIÓN DE NECESIDADES.....	13
5.1	INTRODUCCIÓN.....	13
5.2	DOCUMENTO DE CONCEPTOS DEL SISTEMA.....	13
5.2.1	ALCANCE DEL PROYECTO.....	13
5.2.2	TIPOLOGÍA DE USUARIOS FINALES.....	14
5.2.3	RESTRICCIONES.....	14
6	PUNTOS FUNCIÓN DEL PROYECTO.....	15
6.1	ENTORNO UX (WEB).....	15
6.1.1	OBTENCIÓN PUNTOS FUNCIÓN DEL ENTORNO WEB.....	16
6.2	PLATAFORMA IOT.....	19
6.2.1	OBTENCIÓN PUNTOS FUNCIÓN DE LA PLATAFORMA IOT.....	21
6.3	ENTORNO EDGE.....	23
6.3.1	OBTENCIÓN PUNTOS FUNCIÓN DEL CONCENTRADOR Y DISPOSITIVOS.....	24
6.4	PUNTOS FUNCIÓN TOTALES.....	26
7	COCOMO Y COCOMO II.....	27
7.1	COCOMO.....	27
7.1.1	MODELO BÁSICO.....	28
7.1.2	MODELO INTERMEDIO.....	29
7.1.3	MODELO AVANZADO.....	29
7.2	COCOMO II.....	30
7.3	HERRAMIENTAS.....	30
8	ESTIMACIONES.....	31
8.1	PF AJUSTADOS Y DESPUÉS UTILIZAR COCOMO SIN SUS FACTORES DE AJUSTE.....	35
8.2	PF AJUSTADOS Y DESPUÉS UTILIZAR COCOMOII SIN SUS FACTORES DE AJUSTE.....	36
8.3	PF SIN AJUSTAR Y DESPUÉS UTILIZAR COCOMO CON SUS FACTORES DE AJUSTE.....	37

8.4	PF SIN AJUSTAR Y DESPUÉS UTILIZAR COCOMOII CON SUS FACTORES DE AJUSTE	38
9	CONCLUSIONES.....	39
10	RECURSOS HUMANOS	40
10.1	Entorno Web	40
10.2	Plataforma Iot.....	41
10.3	Entorno Edge.....	44
10.4	Software Zigbee	45
11	PLANIFICACIÓN	46
12	VISIÓN FUTURA.....	48
13	BIBLIOGRAFIA	48
14	ANEXOS	49
14.1	ANEXO I: MADUREZ DE UNA EMPRESA	49
14.2	ANEXO II: TABLAS DE COMPLEJIDAD PUNTOS FUNCIÓN	50
14.3	ANEXO III: TABLA PUNTOS DE FUNCIÓN AJUSTADOS.....	51
14.4	ANEXO IV: FACTORES DE AJUSTE (EM).....	57
14.5	ANEXO V: FORMULARIO SHEF.....	59

TABLA DE ILUSTRACIONES

Ilustración 1: Esquema Entorno proyecto	7
Ilustración 2: Comunicación con plataforma IOT	20
Ilustración 3: Distribución total entorno web	40
Ilustración 4: Organigrama entorno web	40
Ilustración 5: Distribución total plataforma Iot	41
Ilustración 6: Organigrama plataforma Iot	41
Ilustración 7: Organigrama diseño producto (Plat. Iot)	42
Ilustración 8: Organigrama programación (Plat. Iot)	42
Ilustración 9: Organigrama integración y test (Plat, Iot)	43
Ilustración 10: Distribución total entorno Edge	44
Ilustración 11: Organigrama entorno edge	44
Ilustración 12: Distribución total software zigbee	45
Ilustración 13: Organigrama software zigbee	45
Ilustración 14: Cronograma planificación	46

1 INTRODUCCIÓN

La realización de la estimación de un proyecto software es echar un vistazo al futuro en el cual va a existir cierto grado de incertidumbre. Aunque la estimación, es más un arte que una Ciencia, es una actividad importante en la que hay que tener precaución a la hora de realizarse.. Existen técnicas útiles para la estimación de costes de tiempo. Debido a que la estimación es unas de las partes de la planificación de un proyecto, sirve como guía para realizar un buen trabajo.

Al estimar tomamos en cuenta no solo del procedimiento técnico a utilizar en el proyecto, sino que se toma en cuenta los recursos, costos y planificación. El tamaño del proyecto es otro factor importante que puede afectar la precisión de las estimaciones. A medida que el tamaño aumenta, crece rápidamente la interdependencia entre varios elementos del software. La disponibilidad de información histórica es otro elemento que determina el riesgo de la estimación.

Todo proyecto requiere realizar una estimación y una planificación para conocer el alcance que puede tener y así establecer una visión global del proyecto. Cuando se va a realizar un proyecto se suele conocer todo lo que va a estar involucrado en el mismo, lo que hace que se puedan seguir unos pasos para poder realizar toda la operación.

La cosa cambia cuando se habla de realizar la estimación y la planificación de un proyecto software, ya que a la hora de realizar la estimación vamos a trabajar con información que no sabemos *a priori* como se va a desempeñar, por lo que para realizar una buena estimación de software nos puede ayudar a detectar proyectos que no nos conviene abordar o que no son rentables con lo que podemos adelantarnos a conocer si queremos ofertar un proyecto, por eso existen una serie de procedimientos especiales para realizar una estimación y planificación de un proyecto software.

A medida que se vaya desarrollando el proyecto iremos viendo las diferentes formas que hay para realizar el análisis de un proyecto que va a estar centrado en el desarrollo de una plataforma IOT, al cual le vamos a aplicar una serie de métodos y herramientas para analizar las diferentes formas de realizar la planificación del proyecto de varias formas diferentes.

En los últimos años se ha observado un gran aumento por la tecnología relacionada con la domótica. Las tecnologías inalámbricas han adoptado con el paso del tiempo una manera más sencilla y cómoda de utilizar toda clase de dispositivos con el fin de mejorar el confort y las comunicaciones en general. Ésta investigación aborda la tecnología

inalámbrica ZigBee, basada en el estándar 802.15.4 que por su poca introducción al mercado no es muy conocida a pesar de que no es muy reciente.

ZigBee comunica una serie de dispositivos haciendo que trabajen más eficiente entre sí. Es un transmisor y un receptor que usa baja potencia para trabajar y tiene como objetivo las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías. Es ideal para conexiones con diversos tipos de topología, lo que a su vez lo hace más seguro, barato y que no haya ninguna dificultad a la hora de su construcción porque es muy sencilla.

Esta tecnología inalámbrica del futuro que no tiene competencia fuerte con las tecnologías existentes debido a que sus aplicaciones son de automatización de edificios, hogareñas e industriales, especialmente para aplicaciones con usos de sensores.

La tecnología en el hogar es un campo de posibilidades apasionantes y la domótica es una de ellas, es una disciplina técnica, que tiene una faceta social no menos importante. Como primera aproximación afirmaremos que la domótica consiste en introducir info-tecnología en los hogares para mejorar la calidad de vida de sus habitantes y ampliar sus posibilidades de comunicación, automatizando procesos domésticos e intercomunicando tanto estos procesos como los residentes del hogar entre sí y con el exterior. La info-tecnología es la herramienta, las personas sus destinatarios, la satisfacción de determinadas necesidades suyas su objetivo.

En la actualidad no existen muchos dispositivos conectados entre sí por lo que es una buena oportunidad para extender este campo y llevarlo al ámbito del IOT (Internet of Things).

2 DESCRIPCIÓN DEL PROYECTO

El análisis del proyecto que se va a realizar forma parte de un ámbito que se está desarrollando de una manera muy amplia en el mercado de la tecnología en el cual actualmente muchas empresas están apostando por este tipo de proyectos de que se va a hablar a continuación.

El proyecto que se va a analizar está desarrollado en el ámbito de las tecnologías IOT conocido como “*El internet de las cosas*” y consta principalmente de tres partes bien diferenciadas:

- **Entorno UX (Web)**, en el cual se van a poder realizar todas las operaciones que puede hacer el usuario final del producto a desarrollar.
- **La plataforma IOT**, donde se van a gestionar todos los mensajes que transcurren por la plataforma y donde se van a guardar todos los datos que se gestionan en el proyecto, así como todas las configuraciones de cada uno de los usuarios que utilicen el producto.
- **Entorno Edge**, que tratan de todos los dispositivos que se van a instalar en el hogar o edificio del usuario final para la obtención de los diferentes datos del usuario que va a querer controlar.

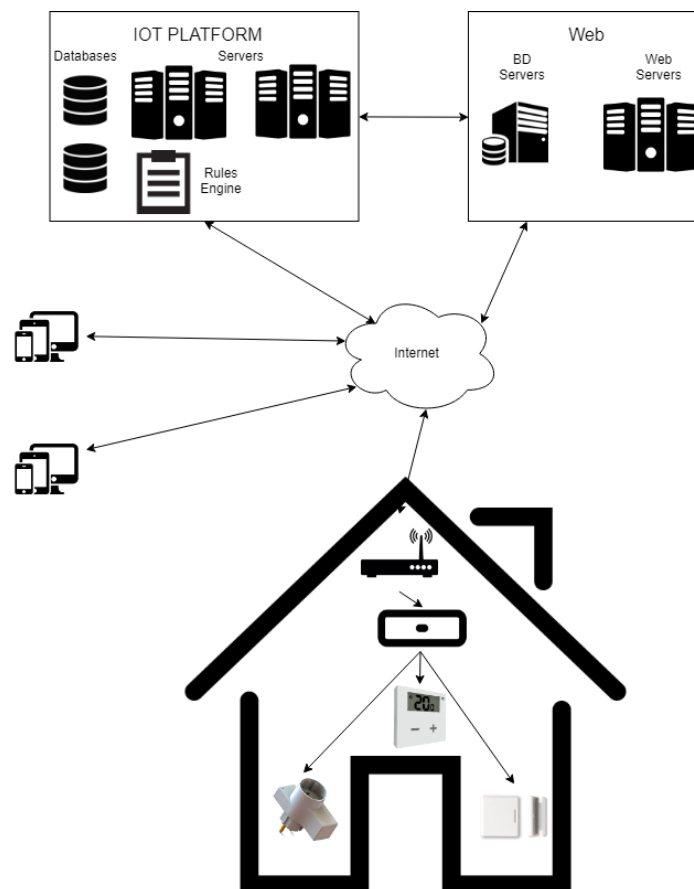


Ilustración 1: Esquema Entorno proyecto

La finalidad del proyecto consiste en realizar el análisis del proyecto mediante la estimación y la planificación enfocándolo de cuatro maneras diferentes para ver las variaciones que se pueden producir a la hora de desarrollarlo de una u otra forma, de modo que se vea la mejor solución para elegir cual puede ser la mejor opción en cuanto a la planificación en tiempo y coste.

Para realizar estos análisis vamos a utilizar las herramientas de COCOMO y COCOMOII al igual que se van a realizar las estimaciones utilizando el método de puntos función (PF).

Se van a realizar cuatro estimaciones independientes, enfocando cada opción de la siguiente manera:

- Puntos de función ajustados y después utilizar COCOMO sin sus factores de ajuste.
- Puntos de función ajustados y después utilizar COCOMO II sin sus factores de ajuste.
- Puntos de función sin ajustar y después utilizar COCOMO con sus factores de ajuste.
- Puntos de función sin ajustar y después utilizar COCOMO II con sus factores de ajuste.

3 OBJETIVOS

- Analizar y comparar métodos y herramientas para la estimación de proyectos.
- Realizar estimación de un proyecto IOT de 4 formas diferentes y poder realizar un análisis de los resultados obtenidos.
- Utilizar diferentes metodologías para el análisis del proyecto.
- Obtener una conclusión de los resultados para poder establecer las ventajas e inconvenientes de cada una de ellas.
- Entender el funcionamiento de las nuevas tecnologías IOT.

4 ESTADO DEL ARTE

Para la mayoría de los proyectos, el factor fundamental del coste es el esfuerzo. El esfuerzo es el número de personas/mes necesarias para desarrollar un proyecto. Es el factor con mayor incertidumbre, que se ve influenciado por factores como la motivación, la experiencia, el nivel de formación y otras características de los miembros del equipo, así como por el tamaño del mismo. Para realizar estimaciones precisas, se han desarrollado técnicas que capturan la relación entre el esfuerzo y las características del personal, los requisitos del proyecto y otros factores que puedan afectar al tiempo, esfuerzo y el coste de desarrollar un proyecto.

Existen tipos de estimación de un proyecto:

Modelo Top-Down: Es una estimación del conjunto completo del proyecto, de tal forma que se va detallando cada parte según se van especificando cada elemento del proyecto, así hasta que la especificación completa es válida para validar el modelo.

Modelo Bottom-Up: Se especifica cada componente del proyecto de forma individual y se van añadiendo más funcionalidades hasta que se llega al modelo general del proyecto.

Existen numerosos métodos de estimación software, que se pueden agrupar en dos grupos principales: Enfoque heurístico o los que siguen un enfoque paramétrico.

Enfoque heurístico

- El método basado en **juicio experto**.

Más comúnmente llamado “a ojo”, y que consiste en preguntar a alguien con experiencia cual es en su opinión la estimación software. Pero todo queda en la propia experiencia del experto por lo que si en algún momento del proyecto, este experto dejara el proyecto nos quedaríamos sin forma de estimar.

- El método por **analogía**.

Es una evolución del anterior, ya que se basa en experiencias documentadas de previos proyectos que se hicieron con anterioridad. El método compara proyectos que hayan sido finalizados anteriormente y que sean similares. Aquí la importante aportación es que disponemos de un método, y de que la experiencia se va guardando en una base de datos.

Enfoque paramétrico

- **COCOMO II** (COConstructive COst MOdel), que estima el esfuerzo (horas hombre) del proyecto. Para estimar el esfuerzo requiere previamente una estimación del tamaño (funcionalidad, líneas de código, etc.).
- **SLIM** (Software Lifecycle Management), de Putnam, que de manera similar contiene un conjunto de fórmulas de estimación software. Estas fórmulas se extrajeron de estudiar grandes bases de datos de proyectos, observando cómo se comportaron las estimaciones software y distribuciones de esfuerzo.

Los dos anteriores sirven principalmente para obtener una estimación del esfuerzo (horas hombre de proyecto) y se basan en que exista previamente un cálculo del tamaño del software a desarrollar. Para determinar el tamaño del software se utilizan principalmente dos unidades de medición: las **líneas de código (LOC)** y los **puntos función (PF)**. Como las líneas de código son poco exactas, lo normal es estimar el tamaño en puntos función, y para ello existen numerosos métodos de estimación software basados en puntos función (FPA de IFPUG, COSMIC-FFP, Puntos Casos de Uso, etc.), que estiman el tamaño funcional de un producto software desde los requisitos.

Otra forma de realizar la gestión de un proyecto es la utilización de lo que se conoce como **metodologías ágiles**.

Una de las características de la gestión de proyectos ágiles es el ser una actividad adaptativa en vez de predictiva. Por lo que los procesos de estimación y planificación en un proyecto ágil sean radicalmente diferentes a los de un proyecto tradicional.

Los proyectos ágiles se enfocan en lo siguiente, obtenido del punto 4 de la bibliografía.

- **“Paquetes de trabajo a precio fijo:** *Todo el proyecto se desglosa en versiones más pequeñas que contribuyen al resultado total del producto. Cada lanzamiento es un paquete de trabajo que tiene un precio en consecuencia. A medida que los paquetes de trabajo se completan, los futuros paquetes de trabajo se estiman nuevamente guiándonos por lo que hemos aprendido del anterior. Así se evita una contingencia innecesaria y permite un nivel de reorganización de prioridades y nuevas/reconsideradas características a ser definidas por el cliente.*
- **Terminación anticipada:** *Esto le permite al cliente terminar el proyecto antes si se ha entregado un gran porcentaje del producto, y no hay más retorno de inversión que deba alcanzarse mediante la retención de un equipo que solo dará ganancias marginales. Esta cláusula está permitida generalmente en cualquier momento y es válida siempre y cuando el equipo del proyecto y el cliente hayan mantenido una fuerte, sincera y estrecha relación de trabajo colaborativo. El beneficio para el cliente es que el proyecto terminará antes, al haber entregado todas las características ventajosas necesarias para hacer el producto factible. A cambio, al proveedor se le paga el 20 por ciento del valor del contrato restante y se compensa el riesgo de retención de personal.*
- **Cambios flexibles:** *El cambio es un tema que corre con fuerza en cuanto a la realización de proyectos ágiles. Esperamos no saber todo cuando tenemos que hacer un producto exitoso desde el principio. Así que promovemos el cambio, basándonos en datos relevantes y comentarios sobre el producto para asegurarnos que el adecuado sea entregado.*
- **Trabajo adicional:** *A lo largo de la vida de un proyecto, pueden ser identificadas más características que no serían alcanzables dentro de un proyecto ya estimado. Ante este escenario, cualquier paquete de trabajo nuevo puede ser agregado al final del proyecto o enfocarse en tiempo y materiales.”*

La planificación y estimación ágil está basada en un número de técnicas que un equipo de desarrollo puede utilizar para ganar más confianza en su tamaño, esfuerzo, duración y costo.

Algunos de los métodos de estimación ágiles son XP y Scrum, entre otros.

La metodología XP consiste en ajustarse a una serie de valores centradas en los requisitos del cliente para lograr un producto de buena calidad y en poco tiempo, centrada en las relaciones interpersonales para lograr el éxito en el desarrollo de software. Esta metodología promueve el trabajo en equipo preocupándose del aprendizaje de los

desarrolladores y estableciendo un buen ambiente de trabajo. Es una metodología apropiada cuando existen unos requisitos imprecisos y cambiantes.

La metodología Scrum es un proceso de buenas prácticas que se aplican de manera regular para trabajar en equipo y así obtener el mejor resultado del proyecto, de tal forma que se realizan entregas parciales del producto final utilizado especialmente en proyectos complejos y con requisitos cambiantes donde la flexibilidad, la innovación y la competitividad son fundamentales.

Se podrían establecer una serie de diferencias entre metodologías tradicionales y ágiles que dan lugar a la comparación de las mismas:

- La metodología tradicional se basa en la planificación de tareas mientras que la ágil realiza la planificación en función de los objetivos de negocio.
- En la planificación ágil se realizan muchas entregas de tal forma que el cliente tiene un feedback constante del estado del proyecto mientras que la tradicional puede que sólo exista la entrega al finalizar el proyecto
- En la planificación ágil se hace partícipe a todo el equipo sobre la planificación de un proyecto.

También es importante destacar el uso de los factores de ajuste que se utilizarán para realizar las estimaciones de los PF y de COCOMO tal y como se detallan en los anexos¹ ya que una vez se han obtenido los PF correspondientes es importante aplicarle estos factores de ajuste que van a depender de los requisitos del sistema y del entorno en el cual se va a desarrollar la producción del sistema para realizar una corrección sobre los PF sin ajustar para poder ser más precisos a la hora de realizar la estimación.

¹ Anexo III

5 IDENTIFICACIÓN DE NECESIDADES

5.1 INTRODUCCIÓN

En esta etapa se definen las actuaciones que se pretenden alcanzar. Además, se establecen los límites del proyecto, los objetivos y usuarios.

En el Documento de Conceptos del Sistema especifica la información obtenida en la etapa de Identificación de Necesidades.

Esta información consiste en:

- **Objetivos del proyecto:** Son objetivos de tipo empresarial y no informático.
- **Alcance del proyecto:** Identificación de las funciones de negocio que pretenden automatizarse.
- **Tipología de usuarios finales:** Conocer el perfil de personas a quien va dirigida la aplicación.
- **Restricciones:** Deben considerarse aquellas restricciones que puedan afectar al plan del proyecto y su desarrollo.

5.2 DOCUMENTO DE CONCEPTOS DEL SISTEMA

5.2.1 ALCANCE DEL PROYECTO

La construcción de este sistema implica desarrollar e implementar las siguientes funciones que se dividen en tres apartados principales.

➤ **Entorno UX (Web):**

En este apartado se desarrolla toda la parte Web que contiene todo lo relacionado con la interacción del usuario final y desde donde el usuario podrá gestionar sus dispositivos y el resto de información

➤ **Plataforma IOT:**

Este apartado gestionará toda la información recopilada del usuario al igual que todas las acciones requeridas para poder comunicar la parte de Front con la parte del concentrador, esto incluye el tratado de todos los datos al igual que la ejecución de las acciones de las que se compone la plataforma.

➤ **Entorno Edge:**

Este apartado está contemplado para el desarrollo de la comunicación de los dispositivos mediante un concentrador principal que interrogará a cada uno de los dispositivos para conocer el estado de cada uno de ellos y enviar los datos que sean necesarios a la plataforma IOT.

5.2.2 TIPOLOGÍA DE USUARIOS FINALES

Los usuarios principales a los que va a ir dirigido este producto no tienen ningún ámbito específico por lo que se entenderá como usuario a cualquier persona que quiera tener el producto en su casa.

5.2.3 RESTRICCIONES

Las restricciones que presenta son las siguientes:

- Gran parte de la información que se va a guardar pertenece a personas físicas y a su entorno, por lo que habrá que tener muy en cuenta la Ley de Protección de datos (LOPD).
- Debido a la complejidad del proyecto será necesario el desarrollo del proyecto para empresas con nivel de madurez no menor de 3 (CMMI 3)²

² Niveles de madurez de una empresa (Anexo I)

6 PUNTOS FUNCIÓN DEL PROYECTO

Para realizar la estimación del proyecto se va a llevar a cabo una secuencia de acciones cuyo objetivo será el de obtener unos resultados intermedios, que servirán como medida parcial o aproximada del proyecto y en función de ellos obtener resultados totales referidos al esfuerzo y tiempo de desarrollo del proyecto.

Dependiendo de las diferentes formas en la que vamos a realizar la estimación del proyecto se van a realizar una serie de operaciones que se irán explicando a medida que se vayan desarrollando cada una de las estimaciones que se han comentado anteriormente en el proyecto.

Tal como hemos comentado anteriormente se va a realizar la estimación mediante Puntos función (PF) sin ajustar como primer paso para realizar la estimación.

6.1 ENTORNO UX (WEB)

Para este apartado se ha definido la utilización de un framework llamado AngularJs, por lo que el desarrollo se realizará en Java para la parte de back y JavaScript para los diferentes elementos que van a formar la web. El contenido que formará la parte web constará de diferentes apartados.

- Pantalla de inicio de sesión de usuarios
- Una pantalla tipo panel de Control donde se podrán visualizar, agrupados por estancias, todos los dispositivos asociados y sus valores de forma fácil para poder tener una visión general del estado de la vivienda, al igual que un widget con la información actual de los datos meteorológicos de la localidad del usuario
- Una pestaña de consumo que contendrá componentes más específicos en el apartado de energía, donde se podrá encontrar gráficas de los diferentes consumos repartidos por horas, días, semanas, meses y años.
- Una pestaña de confort en la que se podrá tener acceso a los diferentes elementos del termostato para poder operar con él. También contendrá gráficas de los valores medios de temperatura.
- Una pestaña de configuración en la que se añadirán los diferentes elementos, como datos personales del usuario, estancias de la casa, equipos que se quieran controlar con los dispositivos, así como un resumen todos los dispositivos asociados.
- Una pestaña que contendrá todo el histórico de eventos que se hayan producido de cada uno de los dispositivos disponibles.

6.1.1 OBTENCIÓN PUNTOS FUNCIÓN DEL ENTORNO WEB

6. 1. 1. 1 INICIO DE SESION

- Usuario (ILF) con 2 DET (usuario, contraseña) = complejidad baja = 7 PF
- Introducción de datos de usuario (EI) con 2 DET (usuario, contraseña) = complejidad baja = 3 PF
- Envío de datos para comprobación de usuario (EO) con 2 DET (usuario, contraseña) = complejidad baja = 4 PF

TOTAL = 14 PF

6. 1. 1. 2 PANEL DE CONTROL

- Dispositivo (ILF) con 5 DET (nombre, id_dispositivo, medida, estancia, timestamp) = complejidad baja = 7 PF
- Widget meteorológico (ILF) con 5 DET (localidad, temperatura máxima del día, temperatura mínima del día, temperatura actual, presión atmosférica) = complejidad baja = 7 PF
- Obtención de datos del dispositivo (EI) (ILF dispositivo) con 5 DET (nombre, id_dispositivo, medida, estancia, timestamp) = complejidad baja = 3 PF
- Acción sobre un dispositivo (EI) (ILF dispositivo) con 3 DET (id_dispositivo, valor, timestamp) = complejidad baja = 3 PF
- Envío de la acción al dispositivo (EO) (ILF dispositivo) con 3 DET (id_dispositivo, valor, timestamp) = complejidad baja = 4 PF
- Obtención de datos meteorológicos (EI) (ILF widget meteorológico) con 5 DET (localidad, temperatura máxima del día, temperatura mínima del día, temperatura actual, presión atmosférica) = complejidad baja = 3 PF

TOTAL = 27 PF

6. 1. 1. 3 PESTAÑA CONSUMO

- Gráfica de consumo (ILF) con 2 DET (fecha y valor) = complejidad baja = 7 PF
- Widget consumo pinza amperimétrica (ILF) con 5 DET (valor, fecha, valor máximo, fecha valor máximo) = complejidad baja = 7 PF
- Obtención de datos gráfica (EQ) (ILF gráfica de consumo) con 5 DET (periodo, id dispositivo, fecha inicio, fecha fin, señal) = complejidad baja = 4 PF
- Obtención de datos del widget (EI) (ILF widget consumo) con 5 DET (valor, fecha, valor máximo, fecha valor máximo) = complejidad baja = 3 PF

TOTAL = 21 PF

6. 1. 1. 4 PESTAÑA CONFORT

- Widget Termostato (ILF) con 4 DET (temp. ambiente, consigna, modo, estado) = complejidad baja = 7 PF
- Gráficas comparativas (ILF) con 4 DET (fecha y valor consumo, fecha y valor temperatura) = complejidad baja = 7 PF
- Obtención de datos del widget termostato (EI) (ILF widget termostato) con 4 DET (temp. ambiente, consigna, modo, estado) = complejidad baja = 3 PF
- Cambio de consigna del termostato (EO) (ILF widget termostato) con 4 DET (id dispositivo, valor de consigna, estado) = complejidad baja = 4 PF
- Obtención de datos de la gráfica (EI) (ILF Gráfica comparativa) con 4 DET (fecha y valor consumo, fecha y valor temperatura) complejidad baja = 3 PF

TOTAL = 24 PF

6. 1. 1. 5 PESTAÑA CONFIGURACIÓN

- Configuración vivienda (ILF) con 3 DET (número de pisos de la vivienda, estancia, equipos) = complejidad baja = 7 PF
- Dispositivo (ILF) con 5 DET (nombre, estancia, equipo, valor, señal, fecha última actualización) = complejidad baja = 7 PF

- Configuración de vivienda (EO) con 3 DET (número de pisos de la vivienda, estancia, equipos) = complejidad baja = 4 PF
- Obtención de datos de los dispositivos (EI) con 5 DET (nombre, estancia, equipo, valor, señal, fecha última actualización) = complejidad baja = 3 PF

TOTAL= 21 PF

6. 1. 1. 6 PESTAÑA HISTORICO DE EVENTOS

- Tabla de eventos (ILF) con 5 DET (Id evento, nombre dispositivo, evento, valor, timestamp) = complejidad baja = 7 PF

- Obtención de todos los eventos (EI) (ILF tabla de eventos) con 5 DET (Id evento, nombre dispositivo, evento, valor, timestamp) = complejidad baja = 3 PF
- Eliminar eventos (EO) (ILF tabla de eventos) con 1 DET (Id evento) = complejidad baja = 4 PF

TOTAL = 14 PF

6. 1. 1. 7 PUNTOS FUNCIÓN TOTALES

Una vez hemos obtenido todos los puntos función de cada uno de los apartados, podemos obtener un resultado.

Tarea	PF
Inicio de sesión	14
Panel de control	27
Pestaña de consumo	21
Pestaña de confort	24
Pestaña de configuración	21
Pestaña de histórico de eventos	14
TOTAL	121

6. 2 PLATAFORMA IOT

Para realizar la estimación de la plataforma IOT, se van a desglosar en diferentes apartados, los cuales se comentan a continuación.

- Desarrollo de la ontología para establecer un intercambio de información entendible entre la parte web y la parte del concentrador.
- Creación de la librería java que será utilizado por ambos entornos para una correcta conexión e interacción con la plataforma.
- Desarrollo de la base de datos en la que se va a guardar toda la información relevante de los dispositivos.
- Desarrollo de la base de datos para el almacenamiento de los datos históricos que se utilizarán para obtener información agrupada de los mismos para diferentes periodos de tiempo.
- Creación de alertas y programas que actuarán sobre los dispositivos y avisarán de anomalías detectadas.

- Gestión de todos los mensajes enviados por el entorno web y el concentrador para la comunicación entre ambos concentradores.

A continuación, se muestra un esquema de los elementos que participan en este apartado.

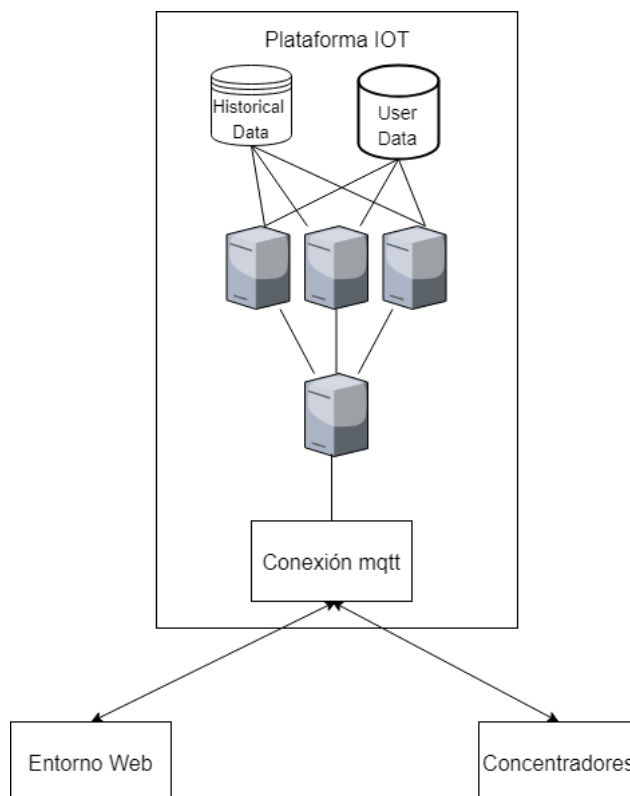


Ilustración 2: Comunicación con plataforma IOT

Debido a la envergadura del apartado de la plataforma IO se ha optado por realizar una estimación heurística mediante el **juicio de un experto**. Ya que anteriormente he realizado proyectos similares, puedo obtener un resultado preciso de los diferentes puntos función diferenciándolos en los diferentes puntos que se muestran en el siguiente apartado.

6.2.1 OBTENCIÓN PUNTOS FUNCIÓN DE LA PLATAFORMA IOT

6. 2. 1. 1 DESARROLLO DE LA ONTOLOGÍA

- Para obtener los puntos función de este apartado se han tenido en cuenta todos los posibles mensajes que se pueden intercambiar, como son los mensajes de Request, Response, Feeds y Events, cada uno utilizado para preguntar por un dato, responder a una pregunta, mandar monitorización de datos y enviar eventos de dispositivos, respectivamente.
- Comprobaciones y verificaciones de que los datos que se envían son coherentes.
- Implementación del proceso de conexión con la plataforma.

Para cada uno de estos puntos se han establecido los siguientes puntos función: **320PF**

6. 2. 1. 2 CREACIÓN LIBRERÍA PARA COMUNICACIÓN CON LA PLATAFORMA

- Implementación de los diferentes mensajes que se intercambiarán entre plataforma y entorno web y entre la plataforma y el entorno de concentradores.
- Desarrollo de la conexión mqtt que se realizará para conectarse a la plataforma.
- Validación de los mensajes enviados.
- Desarrollo de la conexión con la plataforma al igual que la gestión de reconexiones.

Para cada uno de estos puntos se han establecido los siguientes puntos función: **230PF**

6. 2. 1. 3 BASE DE DATOS DE DISPOSITIVOS

- Creación de las tablas necesarias para la gestión de dispositivos.

Para cada uno de estos puntos se han establecido los siguientes puntos función: **100 PF**

6. 2. 1. 4 BASE DE DATOS DE ALMACENAMIENTO DE DATOS HISTÓRICOS

- Creación de las tablas necesarias para el almacenamiento de todos los datos de los dispositivos.
- Gestión de los datos para la obtención de históricos de los diferentes datos de los que disponga un usuario.

Para cada uno de estos puntos se han establecido los siguientes puntos función: **120PF**

6. 2. 1. 5 CREACIÓN DE ALERTAS Y PROGRAMAS

- Creación de toda la lógica necesaria para la creación y ejecución de las alertas y programas que pueda tener un usuario para sus dispositivos.

Para cada uno de estos puntos se han establecido los siguientes puntos función: **150PF**

6. 2. 1. 6 GESTIÓN DE MENSAJES DE INTERCAMBIO ENTRE LOS DIFERENTES APARTADOS

- Creación de la pasarela de comunicación necesaria para realizar la comunicación entre los tres componentes.

Para cada uno de estos puntos se han establecido los siguientes puntos función: **98 PF**

6. 2. 1. 7 PUNTOS FUNCIÓN TOTALES

Una vez hemos obtenido todos los puntos función de cada uno de los apartados, podemos obtener un resultado.

Tarea	PF
Desarrollo de la ontología	320
Creación librería para comunicación con la plataforma	230
Base de datos de dispositivos	100
Bases de datos de datos históricos	120
Creación de alertas y programas	150
Gestión de mensajes de intercambio	98
TOTAL	1018

6. 3 ENTORNO EDGE

Para realizar la estimación del apartado Edge, se van a desglosar en diferentes apartados, los cuales se comentan a continuación.

- Creación software para la comunicación con los diferentes dispositivos utilizando comunicación zigbee.
- Creación de software para realizar la comunicación con la plataforma IOT.
- Gestión de los mensajes con la plataforma IOT
- Envío periódico de datos de los diferentes dispositivos a la plataforma IOT
- Gestión de eventos de los dispositivos

6.3.1 OBTENCIÓN PUNTOS FUNCIÓN DEL CONCENTRADOR Y DISPOSITIVOS

6.3.1.1 SOFTWARE ZIGBEE

Referente a este apartado y debido a la complejidad del mismo se ha optado por obtener el valor de PF y realizar una estimación heurística mediante el **juicio de un experto**.

Actualmente ya he trabajado con el desarrollo de diferentes protocolos de comunicaciones, tales como modbus, zigbee y opcua y por lo tanto puedo establecer de manera precisa los puntos función que se pueden obtener para realizar este apartado del proyecto, lo que no deja un resultado total de **447 PF**, que son exactamente los que se obtuve anteriormente en proyectos anteriores.

6.3.1.2 COMUNICACIÓN CON LA PLATAFORMA IOT

- Datos de conexión (ILF) con 5 DET (url, puerto, usuario, contraseña, tópico) = complejidad baja = 7 PF

- Petición de conexión (EO) con 5 DET (url, puerto, usuario, contraseña, tópico) = complejidad baja = 4 PF

- Respuesta de conexión (EI) con 2 DET (estado, token de conexión) = complejidad baja = 3 PF

TOTAL = 14 PF

6.3.1.3 GESTIÓN DE MENSAJES CON LA PLATAFORMA IOT

- Mensaje de entrada y salida (ILF) con 13 DET (commandId, timestamp, assetId, assetSource, targetKpInstance, requestId, instancekp, signalId, type, unit, signal, value, description) = complejidad baja = 7 PF

- Petición de datos de un dispositivo o actuación (EI) (ILF Mensaje de entrada y salida) con 13 DET = (commandId, timestamp, assetId, assetSource, targetKpInstance, requestId, instancekp, signalId, type, unit, signal, value, description) = complejidad baja = 3 PF

- Recepción de mensaje (EO) (ILF Mensaje de entrada y salida) con 13 DET = (commandId, timestamp, assetId, assetSource, targetKpInstance, requestId, instancekp, signalId, type, unit, signal, value, description) = complejidad baja = 4 PF

TOTAL = 14 PF

6.3.1.4 ENVÍO PERIÓDICO DE DATOS A LA PLATAFORMA IOT

- Datos de dispositivo (ILF) con 7 DET (assetSource, timeSource, measure, assetId, unit, equipment, signal) = complejidad baja = 7 PF

- Envío de datos (EO) (ILF datos del dispositivo) con 7 DET (assetSource, timeSource, measure, assetId, unit, equipment, signal) = complejidad baja = 4 PF

TOTAL = 11 PF

6.3.1.5 GESTIÓN DE EVENTOS DE LOS DISPOSITIVOS

- Datos evento (ILF) con 7 DET (commandId, timestamp, assetId, instancekp, assetSource, status, statusprevious) = complejidad baja = 7 PF

- Envío de evento (EO) (ILF datos evento) con 7 DET (commandId, timestamp, assetId, instancekp, assetSource, status, statusprevious) = complejidad baja = 4 PF

TOTAL = 11 PF

6. 3. 1. 6 PUNTOS FUNCIÓN TOTALES

Una vez hemos obtenido todos los puntos función de cada uno de los apartados, podemos obtener un resultado.

Tarea	PF
Software zigbee	447
Comunicación con plataforma IOT	14
Gestión de mensajes con plataforma IOT	14
Envío periódico de datos a plataforma IOT	11
Gestión de eventos de los dispositivos	11
TOTAL	497

6. 4 PUNTOS FUNCIÓN TOTALES

Después de haber realizado la estimación mediante puntos función de cada apartado hemos obtenido un el siguiente resultado.

Apartado	PF
Entorno UX (Web)	121
Plataforma IOT	1018
Entorno Edge	497
TOTAL	1636

7 COCOMO Y COCOMO II

Antes de empezar a realizar las estimaciones con COCOMO y COCOMO II vamos a realizar una pequeña introducción a cada uno de ellos para comprender mejor las diferencias que tiene cada uno de ellos.

7.1 COCOMO

COCOMO (COConstructive COSt MOdel) fue desarrollado por Barry W. Boehm a finales de los años 70 y comienzos de los 80 y es un modelo utilizado para la estimación de costes de software.

Este modelo incluye tres niveles de desarrollo:

- **Orgánico:** Son proyectos sencillos de no más de 50 KSLOC³ del cual se tiene experiencia en proyectos similares y se encuentra en entornos estables
- **Semi-acoplado:** Son proyecto de media dificultad que no llegan a las 300 KSLOC y cuyo conocimiento de proyectos similares es variable
- **Empotrado:** Son proyectos complejos de los que apenas se tiene experiencia y se encuentran en el apartado de innovación técnica al igual que se trabajan con requisitos muy restrictivos.

Por otro lado, existen diferentes modelos dentro de COCOMO.

³ KSLOC: unidad de medida. 1 KSLOC equivalen a mil líneas de código.

7.1.1 MODELO BÁSICO

Este modelo se utiliza para estimar el esfuerzo y el tiempo utilizado para realizar un proyecto software usando dos variables predictivas denominadas factores de costos, el tamaño del software y el modo de desarrollo, para cada uno de estos factores se utilizan las siguientes ecuaciones.

$$PM = A \times (KSLOC)^B$$

$$TDEV = C \times (PM)^D$$

PM: esfuerzo estimado (meses-hombre)

TDEV: meses de trabajo necesarios para ejecutar el proyecto

A y B: coeficientes que dependen del modelo de desarrollo que se utilice.

C y D: coeficientes que dependen del modelo de desarrollo que se utilice.

KSLOC: tamaño del software en miles de líneas de código.

En la siguiente tabla se pueden identificar los valores de los coeficientes dependiendo del modelo de desarrollo que se vaya a utilizar.

Modo de Desarrollo	Esfuerzo	Cronograma
Orgánico	$PM = 2.4 \times (KSLOC)^{1.05}$	$TDEV = 2.5 \times (PM)^{0.38}$
Semi-acoplado	$PM = 3.0 \times (KSLOC)^{1.12}$	$TDEV = 2.5 \times (PM)^{0.35}$
Empotrado	$PM = 3.6 \times (KSLOC)^{1.20}$	$TDEV = 2.5 \times (PM)^{0.32}$

Este método es bueno utilizarlo para una estimación rápida y temprana, pero tiene una precisión limitada.

7.1.2 MODELO INTERMEDIO

Aparte de obtener las líneas de código del proyecto incluye 15 modificadores para mejorar la precisión de la estimación. (Véase Anexo 4)

Para realizar el proceso de estimación de esfuerzo con este modelo se seguirían los siguientes pasos:

- Calcular el *esfuerzo nominal* (PM) igual en el modelo básico.
- Se determina el *factor de ajuste* utilizando la siguiente ecuación.

$$EAF = \prod_{i=1}^{15} EM_i$$

En la siguiente tabla se pueden identificar los valores de los coeficientes dependiendo del modelo de desarrollo que se vaya a utilizar.

Modo de Desarrollo	Esfuerzo	Esfuerzo ajustado
Orgánico	$PM_{\text{nominal}} = 3.2 \times (\text{KSLOC})^{1.05}$	$PM = 3.2 \times EAF \times (\text{KSLOC})^{1.05}$
Semi-acoplado	$PM_{\text{nominal}} = 3.0 \times (\text{KSLOC})^{1.12}$	$PM = 3.0 \times EAF \times (\text{KSLOC})^{1.12}$
Empotrado	$PM_{\text{nominal}} = 2.8 \times (\text{KSLOC})^{1.20}$	$PM = 2.8 \times EAF \times (\text{KSLOC})^{1.20}$

Este método tiene dos inconvenientes a la hora de utilizarlo con un software que contenga un gran número de componentes.

- La estimación del esfuerzo por fases resulta imprecisa
- No es práctico si el software contiene un gran número de componentes.

7.1.3 MODELO AVANZADO

Este modelo incluye el modelo intermedio añadiéndole además el impacto de cada conductor de coste en las distintas fases del desarrollo. Provee un conjunto de fases para estimar el esfuerzo ya sea por módulos o por el proyecto entero.

Para guardar los datos necesarios se emplean unos formularios de tipo SHEF⁴. Un ejemplo de formulario se puede ver en el Anexo V.

⁴ Software Hierarchy Estimating Form

7.2 COCOMO II

COMOMO II es un modelo que se utiliza para estimar el esfuerzo, el costo y la planificación de un proyecto o actividad de software, al igual que proveer de un marco analítico cuantitativo y un conjunto de herramientas que evaluarán el impacto sobre los costos y tiempos en las diferentes etapas del ciclo de vida del proyecto.

Se compone de tres submodelos, cada uno de ellos se utiliza en una fase del proyecto.

- **Composición de la aplicación:** Es utilizado durante las primeras etapas donde se realizan el prototipado de las interfaces de usuario, interacción del sistema, evaluación del sistema y la evaluación de la madurez de la tecnología.

- **Diseño temprano:** Se utiliza una vez que ya se conocen los requisitos del sistema y se ha establecido la arquitectura básica del sistema.

- **Post-Arquitectura:** Utilizada durante la construcción del software. En este modelo se utiliza:

- Puntos función.

- Un conjunto de 17 atributos denominados factores de costo (Véase Anexo 4) que permiten entender mejor características del proyecto referentes al personal, plataforma de desarrollo, etc. que pueden influir en los costos del proyecto.

7.3 HERRAMIENTAS

Una vez explicado cómo funciona tanto COCOMO como COCOMO II podemos utilizar diferentes herramientas de las que nos podemos ayudar para realizar las estimaciones. Algunas de las herramientas que existe son:

- **COCOMO® II.2000.0 software:** Es un software que ofrece la implementación de COCOMO II con los modelos de Post-Arquitectura y Early Access.

https://csse.usc.edu/csse/research/cocomoii/cocomo_downloads.htm

- **Constructive Cost Model:** Es una Aplicación web desde la cual se puede realizar una estimación mediante COCOMO II

<https://csse.usc.edu/tools/COCOMOII.php>

Para nuestro proyecto vamos a utilizar **COCOMO® II.2000.0 software** debido a que es el modelo más completo para poder obtener toda la información que necesitamos y también por la facilidad que uso que proporciona la herramienta a la hora de añadir y obtener los resultados que queremos.

8 ESTIMACIONES

Para realizar los puntos función ajustados se va a utilizar la siguiente función:

$$VAF = (GTI * 0.01) + 0.65$$

$$PFA = PF * VAF$$

Siendo:

VAF = Factor de ajuste (Value Adjustment Factor)

PFA = puntos función ajustados.

GTI = suma de los grados de cada una de las 14 características (Anexo III)

- **Comunicación de datos: (Puntos: 5)**

En el proyecto existen diferentes protocolos para la comunicación de datos (zigbee, mqtt, http) y con interfaz para la comunicación con el exterior

- **Procesamiento distribuido de datos: (Puntos: 5)**

Existen procesos distribuidos que se ejecutan tanto en la parte Edge (obtención de información de dispositivos, como en la plataforma (análisis de datos y comandos de ejecución)

- **Rendimiento: (Puntos: 2)**

Existe un proceso en el cual cada día se tiene que realizar un proceso batch que unifique los datos obtenidos del día.

- **Configuraciones fuertemente utilizadas: (Puntos: 1)**

No existe restricción al respecto

- **Frecuencia de transacciones: (Puntos: 2)**

Se prevé ciertos momentos más propicios para que exista un mayor volumen de peticiones a la plataforma debido a los hábitos de la sociedad actual

- **Entradas de datos on-line: (Puntos: 5)**

Se trata de un proceso totalmente interactivo en la que el usuario va a poder obtener la información totalmente actualizada bajo demanda

- **Eficiencia del usuario final: (Puntos: 2)**

Los elementos asociados al usuario final serán: Menús, Uso del ratón, Selección mediante cursor de datos de la pantalla, Ventanas de "pop-up", Aplicación bilingüe.

- **Actualizaciones online: (Puntos: 3)**

Se realizarán actualizaciones de ficheros tanto en el nivel de Edge como en ficheros de la Plataforma IOT

- **Procesamiento complejo: (Puntos: 4)**

El procesamiento cuenta con las siguientes características: Se incorporarán sistemas de seguridad y control, Proceso con lógica compleja, Especificado muchas excepciones, consecuencia de transacciones incompletas, que deberán tratarse, Manejar múltiples dispositivos de entrada / salida.

- **Reusabilidad: (Puntos: 3)**

Se prevé que por lo menos el 10% de código se puede utilizar para más de un usuario

- **Facilidad de instalación: (Puntos: 1)**

No se requiere conversión alguna para una nueva instalación

- **Facilidad de operación: (Puntos: 3)**

Existen Procesos de arranque, back-up y recuperación con o sin intervención del operador y manejo de cintas u otros dispositivos manual o automático.

- **Instalación en distintos lugares: (Puntos: 1)**

La instalación se realizará en un único lugar, por cada usuario que utilice el servicio

- **Facilidad de cambio: (Puntos: 3)**

Existe un procedimiento por el cual se estudia las causas del posible cambio de dispositivos.

A continuación, se muestra un resumen de los resultados.

CARACTERÍSTICA	Valor
Comunicación de datos	5
Procesamiento distribuido de datos	5
Rendimiento	2
Configuraciones fuertemente utilizadas	1
Frecuencia de transacciones	2
Entrada de datos on- line	5
Eficiencia del usuario final	2
Actualizaciones Online	3
Procesamiento complejo	4
Reusabilidad	3
Facilidad de instalación	1
Facilidad de operación	3
Instalación en distintos lugares	1
Facilidad de cambio	3
TOTAL	40

$$VAF = (40 * 0.01) + 0.65 = 1.05$$

$$PFA = 1636 * 1.05 = 1717,8$$

PFSA= 1636

PFA = 1717,8

8. 1 PF AJUSTADOS Y DESPUÉS UTILIZAR COCOMO SIN SUS FACTORES DE AJUSTE

Una vez obtenidos los puntos función de cada una de las tres fases del proyecto en el apartado 6 se va a aplicar COCOMO para realizar la estimación de esfuerzo y coste sin aplicar los factores de ajuste.

Al tratarse de un proyecto de 1717.8 PF, para simplificar se ha decidido realizar la transformación como si todo el software fuera hecho en Java lo que da un total de 91054 Líneas de código o 91 KSLOC lo que supone que el modelo a seguir de COCOMO será el **semi-acoplado**.

$$PM = A \times (KSLOC)^B = 3,0 \times (91)^{1,12} = 669,080 \text{ meses/persona}$$

$$TDEV = Cx(PM)^D = 2,5x(669,080)^{0,35} = 24,37 \text{ meses}$$

8.2 PF AJUSTADOS Y DESPUÉS UTILIZAR COCOMOII SIN SUS FACTORES DE AJUSTE

En esta ocasión se va a aplicar COCOMO II para obtener la estimación de esfuerzo y el coste asociado sin utilizar sus factores de ajuste.

Los PF ajustados obtenidos son de 1717,8 que es lo que vamos a utilizar para realizar el cálculo en COCOMO II.

The screenshot shows the USC-COCOMO II.2000.4 application window. The project name is 'IOT Pj' and the scale factor is 18.97. The development model is set to 'Post Architecture'. The main table lists four modules: Entorno Web, Plataforma Iot, Entorno Edge, and Soft. Zigbee. Each module has associated values for Module Size, Labor Rate, EAF, Language, NCM Effort, EST Effort, PROD, COST, INST COST, Staff, and RISK. A summary table at the bottom provides estimated effort, schedule, and cost for three scenarios: Optimistic, Most Likely, and Pessimistic. The total lines of code are 86708 and the project duration is 152.00 hours per person-month.

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	Language	NCM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	Entorno Web	F:6413	1000.00	1.00	JAVA	29.4	29.4	218.0	29419.35	4.6	1.2	0.0
	Plataforma Iot	F:53954	1000.00	1.00	JAVA	247.5	247.5	218.0	247511.55	4.6	10.1	0.0
	Entorno Edge	F:2650	1000.00	1.00	JAVA	12.2	12.2	218.0	12156.76	4.6	0.5	0.0
	Soft. Zigbee	F:23691	1000.00	1.00	JAVA	108.7	108.7	218.0	108681.40	4.6	4.4	0.0

	Estimated	Effort Sched	PROD	COST	INST	Staff	RISK
Total Lines of Code:	86708						
Hours/PM:	152.00						
Optimistic	318.2	22.9	272.5	318215.25	3.7	13.9	
Most Likely	397.8	24.6	218.0	397769.06	4.6	16.2	0.0
Pessimistic	497.2	26.4	174.4	497211.32	5.7	18.8	

SITE: Multisite Development

Obtendríamos un total de 86708 líneas de código y realizaríamos el proyecto en un periodo de 24,6 meses.

8. 3 PF SIN AJUSTAR Y DESPUÉS UTILIZAR COCOMO CON SUS FACTORES DE AJUSTE

En este caso vamos a utilizar los PF sin ajustar, que son 1636 y los vamos a trasladar a COCOMO utilizando sus factores de ajuste obteniendo la siguiente información.

Los factores de ajuste a los que se les ha cambiado el valor han sido los siguientes: (el resto de ellos se quedarían con el valor nominal 1).

RELY - LO (0,88): En el caso de que haya un error del software no supondrá un problema ya que es fácilmente recuperable.

CPLX - HI (1,15): Existen varios recursos que son necesarios para la obtención de la información, Cuyos datos se obtienen de tratar la información de manera más o menos compleja.

TIME - HI (1,11): Existe una restricción bastante alta en el tiempo que tiene el software para responder con la información que se pide ya que se requiere de actuaciones relativamente rápidas

TOOL - LOW (1,1): El uso de herramientas específicas no es un impedimento ya que se utilizan las que se encuentran actualmente en el mercado.

SCED - LOW (1,1): Contamos con programadores experimentados lo que acelerará el desarrollo del software requerido.

$$EAF = \prod_{i=1}^{15} EM_i = 0,88 * 1,15 * 1,11 * 1,1 * 1,1 = 1,359$$

$$PM = 3.0 \times EAF \times (KSLOC)^{1.12} = 3,0 * 1,359 * (91)^{1.12} = 637,48 \text{ meses/persona}$$

$$TDEV = Cx(PM)^D = 2,5x(637,48)^{0.35} = 23,96 \text{ meses}$$

8.4 PF SIN AJUSTAR Y DESPUÉS UTILIZAR COCOMOII CON SUS FACTORES DE AJUSTE

En este caso vamos a utilizar los PF sin ajustar, que son 1636 y los vamos a trasladar a COCOMO II utilizando sus factores de ajuste obteniendo la siguiente información.

base + Incr % = rating

Product: RELY DATA DOCU CPLX RUSE
 base LO NOM NOM HI NOM
 Incr% 0% 0% 0% 0% 0%

Platform: TIME STOR PVOL
 base HI NOM NOM
 Incr% 0% 0% 0%

Personnel: ACAP PCAP PCON APEX LTEX PLEX
 base NOM NOM NOM NOM NOM
 Incr% 0% 0% 0% 0% 0%

Project: TOOL SITE
 base LO LO
 Incr% 0% 0%

User: USR1 USR2
 base NOM NOM
 Incr% 0% 0%

EAF is also affected by Schedule
 EAF: 1.42

OK Cancel Help

Project Name: IOT Pj Scale Factor: 18.97 Schedule

Project Notes Development Model: Post Architecture

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	Entorno Web	F:6413	1000.00	1.42	JAVA	29.4	41.8	153.6	41762.14	6.5	1.5	0.0
	Plataforma Iot	F:53954	1000.00	1.42	JAVA	247.5	351.4	153.6	351354.19	6.5	12.8	0.0
	Entorno Edge	F:2650	1000.00	1.42	JAVA	12.2	17.3	153.6	17257.08	6.5	0.6	0.0
	Soft. Zigbee	F:23691	1000.00	1.42	JAVA	108.7	154.3	153.6	154278.31	6.5	5.6	0.0

Total Lines of Code:	86708	Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Hours/PM:	152.00	Optimistic	451.7	25.6	192.0	451721.38	5.2	17.6	
		Most Likely	564.7	27.5	153.6	564651.73	6.5	20.5	0.0
		Pessimistic	705.8	29.5	122.8	705814.66	8.1	23.9	

Obtendríamos un total de 86708 líneas de código y realizaríamos el proyecto en un periodo de 27,5 meses.

9 CONCLUSIONES

Una vez que se han realizado los diferentes tipos de estimaciones se va a realizar un análisis de los resultados describiendo unas pautas realizando las siguientes diferencias.

- Diferencias entre usar los factores de ajuste de PF y los de COCOMO
- Diferencias entre usar COCOMO y COCOMO II

Antes de empezar con las conclusiones vamos a poner en claro los resultados que hemos obtenido después de realizar las estimaciones.

	PFSA	PFA	Tiempo (meses)
PFA + COCOMO SIN FA	-	1717,8	24,77
PFA + COCOMO II SIN FA	-	1717,8	24,6
PF + COCOMO CON FA	1636	-	23,96
PF + COCOMO II CON FA	1636	-	27,5

Según podemos observar con los resultados obtenidos las diferencias de utilizar los PFA y los FA de COCOMO o COCOMO II no son prácticamente apreciables a la hora de obtener los resultados de la estimación en tres de las cuatro comparaciones que hemos hecho.

En el caso de la estimación realizada utilizando los PF sin ajustar y después utilizar COCOMO II con sus FA vemos que existe una diferencia notable en cuanto al tiempo se refiere de unos 3 meses. Esto es debido a que los factores de ajuste de COCOMO II son mucho más detallados a la hora de modificar los valores de tiempo para realizar el proyecto.

Debido a este cambio tanto los recursos que se van a utilizar como la planificación que se va a realizar va a estar basada en la última estimación realizada de 27,5 meses.

10 RECURSOS HUMANOS

Para realizar este apartado vamos a separar cada uno de los módulos analizados y vamos a representar y explicar cada uno de los integrantes que van a formar el equipo en cada una de las fases

10.1 Entorno Web

Estos son los datos obtenidos en COCOMO II:

Overall Phase Distribution					
MODULE	Entorno Web				
SLOC	6413				
TOTAL EFFORT	41.762 Person Months				
	PCNT	EFFORT (PM)	PCNT	SCHEDULE	Staff
Plans And Requirements	7.000	2.923	21.140	5.817	0.503
Product Design	17.000	7.100	26.570	7.311	0.971
Programming	56.290	23.508	45.721	12.580	1.869
- Detailed Design	24.430	10.203	----	----	----
- Code and Unit Test	31.860	13.306	----	----	----
Integration and Test	26.710	11.155	27.710	7.624	1.463

Ilustración 3: Distribución total entorno web

Al obtener los resultados de este apartado se ha distribuido el organigrama del módulo de la siguiente manera:

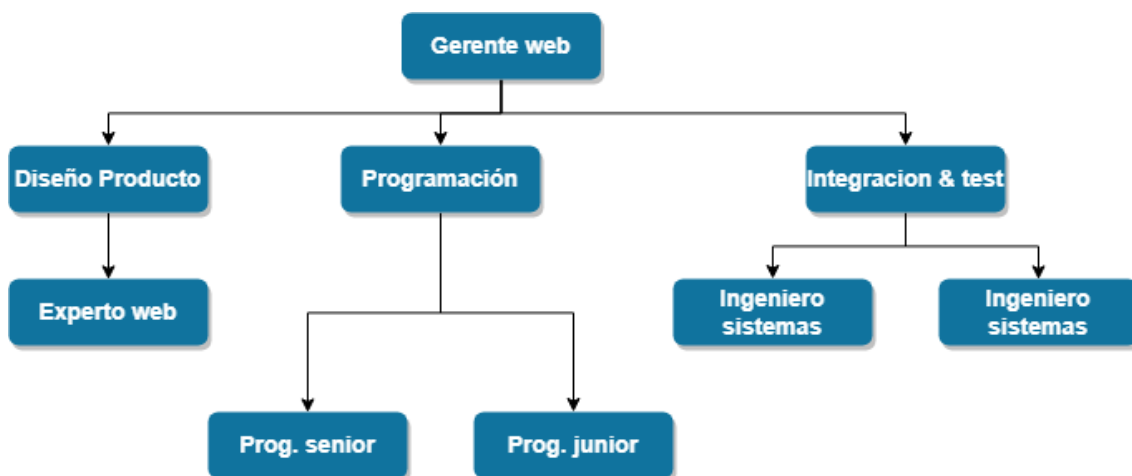


Ilustración 4: Organigrama entorno web

Realizando cada uno de los cargos las siguientes funciones:

- **Gerente web:** encargado de realizar los requerimientos y de coordinar el equipo de trabajo de esta parte del proyecto.
- **Experto web:** responsable del diseño de las páginas web.
- **Programadores:** Encargados de realizar el desarrollo software.
- **Ingenieros de sistemas:** Encargados de realizar las pruebas de integración y test.

10. 2 Plataforma Iot

Estos son los datos obtenidos en COCOMO II

Overall Phase Distribution					
MODULE	Plataforma Iot				
SLOC	53954				
TOTAL EFFORT	351.354 Person Months				
	PCNT	EFFORT (PM)	PCNT	SCHEDULE	Staff
Plans And Requirements	7.000	24.595	21.140	5.817	4.228
Product Design	17.000	59.730	26.570	7.311	8.170
Programming	56.290	197.779	45.721	12.580	15.722
- Detailed Design	24.430	85.836	----	----	----
- Code and Unit Test	31.860	111.942	----	----	----
Integration and Test	26.710	93.845	27.710	7.624	12.309

Ilustración 5: Distribución total plataforma Iot

Al obtener los resultados de este apartado se ha distribuido el organigrama del módulo de la siguiente manera:

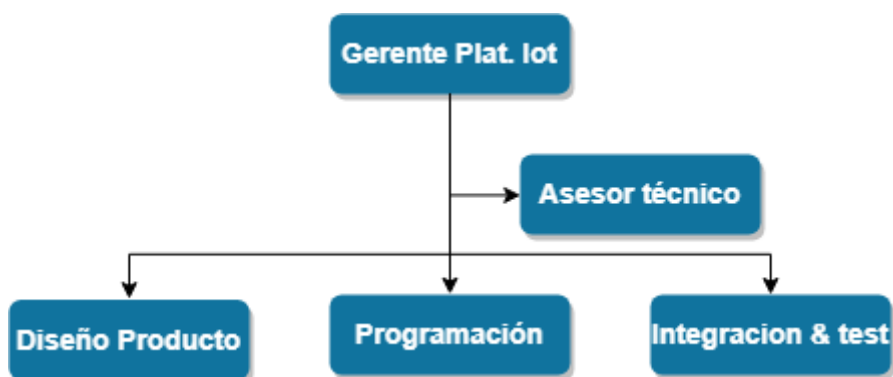


Ilustración 6: Organigrama plataforma Iot

Debido a la magnitud del organigrama se ha optado por separarlo en diferentes organigramas que mostramos a continuación.

1. DISEÑO DEL PRODUCTO

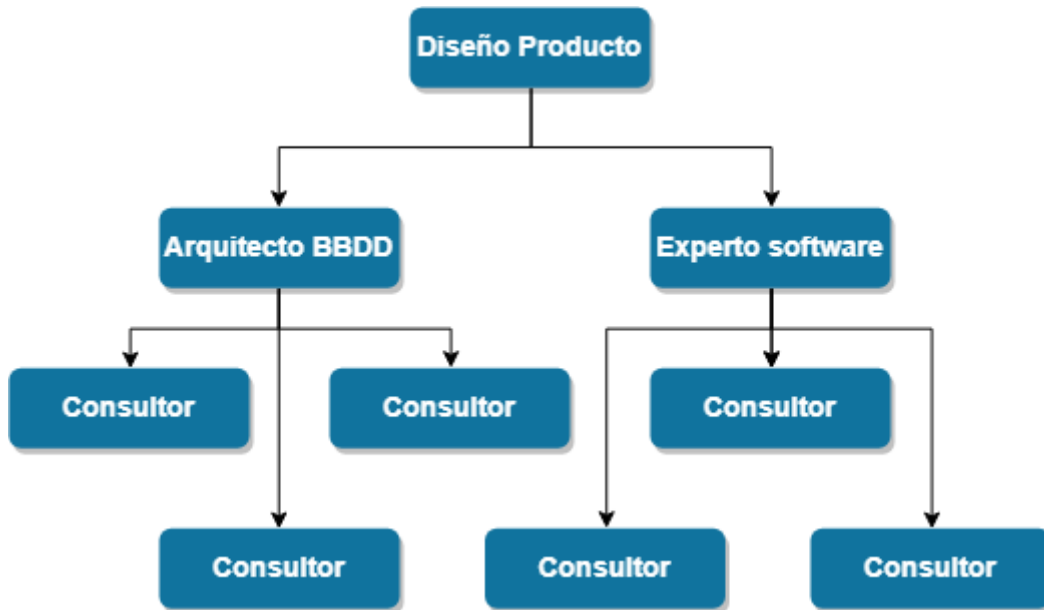


Ilustración 7: Organigrama diseño producto (Plat. Iot)

2. PROGRAMACIÓN

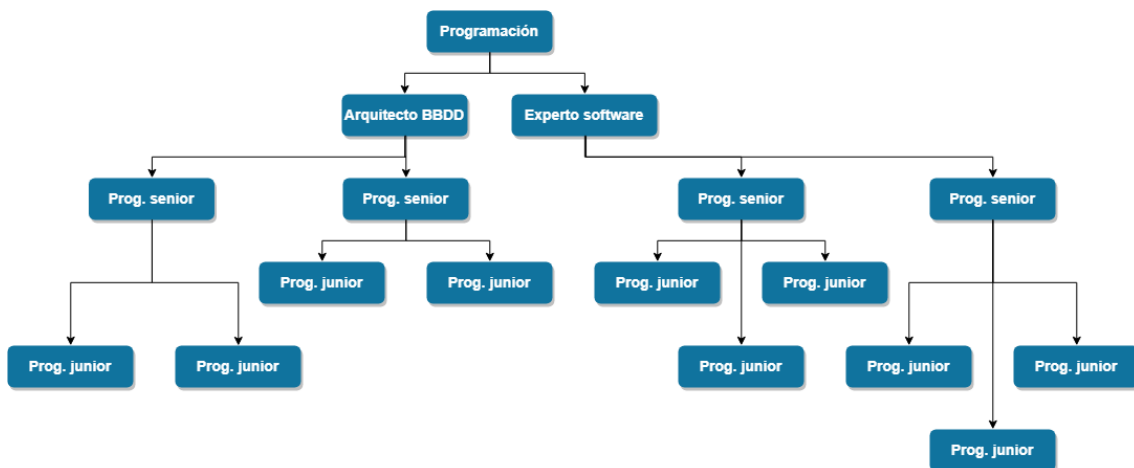


Ilustración 8: Organigrama programación (Plat. Iot)

3. INTEGRACIÓN Y TEST

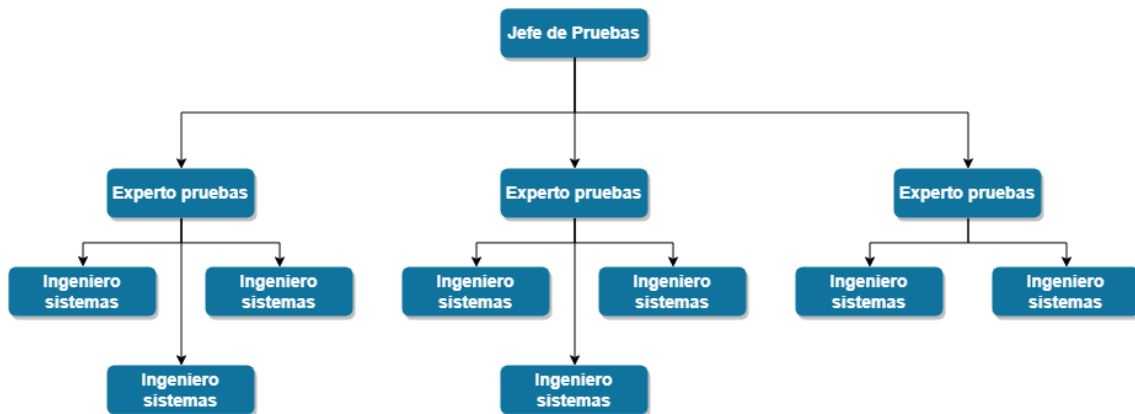


Ilustración 9: Organigrama integración y test (Plat, Iot)

Realizando cada uno de los cargos las siguientes funciones:

- **Gerente plataforma Iot;** encargado de realizar los requerimientos y de coordinar el equipo de trabajo de esta parte del proyecto.
- **Asesor técnico:** dará soporte al gerente para cubrir los aspectos más técnicos del proyecto.
- **Arquitecto BBDD:** responsable del diseño y desarrollo de todas las BBDD necesarias para el proyecto.
- **Experto:** responsable de todos los apartados de comunicación en la plataforma al igual que la gestión de los datos.
- **Consultores:** encargados de diseñar las BBDD, las comunicaciones de la plataforma y gestión de los datos del proyecto.
- **Ingeniero sistemas Senior:** responsable del desarrollo del apartado de comunicaciones y gestión de datos.
- **Programadores:** encargados de realizar el desarrollo del apartado de comunicaciones y gestión de datos.
- **Jefe de pruebas:** responsable del equipo de pruebas de integración y test.
- **Experto pruebas:** responsable de una parte del equipo de pruebas de integración y test.
- **Ingenieros de sistemas:** encargados de realizar las pruebas de integración y test.

10.3 Entorno Edge

Estos son los datos obtenidos en COCOMO II

Overall Phase Distribution					
MODULE	Entorno Edge				
SLOC	2650				
TOTAL EFFORT	17.257 Person Months				
	PCNT	EFFORT (PM)	PCNT	SCHEDULE	Staff
Plans And Requirements	7.000	1.208	21.140	5.817	0.208
Product Design	17.000	2.934	26.570	7.311	0.401
Programming	56.290	9.714	45.721	12.580	0.772
- Detailed Design	24.430	4.216	----	----	----
- Code and Unit Test	31.860	5.498	----	----	----
Integration and Test	26.710	4.609	27.710	7.624	0.605

Ilustración 10: Distribución total entorno Edge

Al obtener los resultados de este apartado se ha distribuido el organigrama del módulo de la siguiente manera:

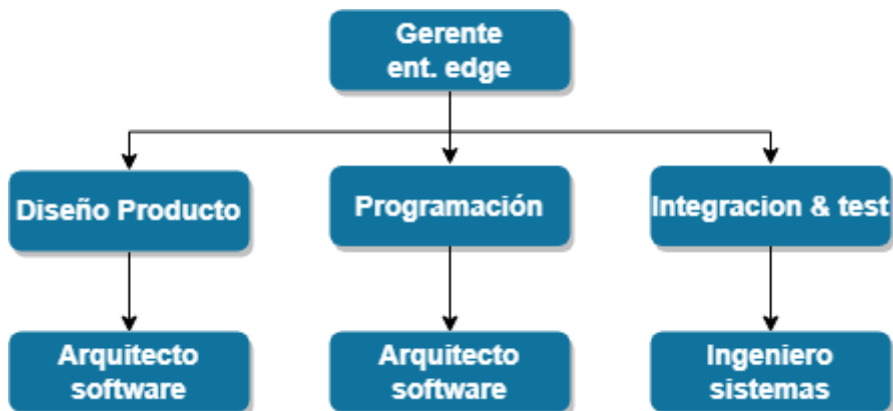


Ilustración 11: Organigrama entorno edge

Realizando cada uno de los cargos las siguientes funciones:

- **Gerente entorno edge:** encargado de realizar los requerimientos y de coordinar el equipo de trabajo de esta parte del proyecto.
- **Arquitecto de software:** responsable del diseño o desarrollo del software.
- **Ingeniero sistemas:** encargado de realizar las pruebas de integración y test.

10. 4 Software Zigbee

Estos son los datos obtenidos en COCOMO II

Overall Phase Distribution					
MODULE	Soft. Zigbee				
SLOC	23691				
TOTAL EFFORT	154.278 Person Months				
	PCNT	EFFORT (PM)	PCNT	SCHEDULE	Staff
Plans And Requirements	7.000	10.799	21.140	5.817	1.857
Product Design	17.000	26.227	26.570	7.311	3.588
Programming	56.290	86.844	45.721	12.580	6.903
- Detailed Design	24.430	37.690	----	----	----
- Code and Unit Test	31.860	49.153	----	----	----
Integration and Test	26.710	41.207	27.710	7.624	5.405

Ilustración 12: Distribución total software zigbee

Al obtener los resultados de este apartado se ha distribuido el organigrama del módulo de la siguiente manera:

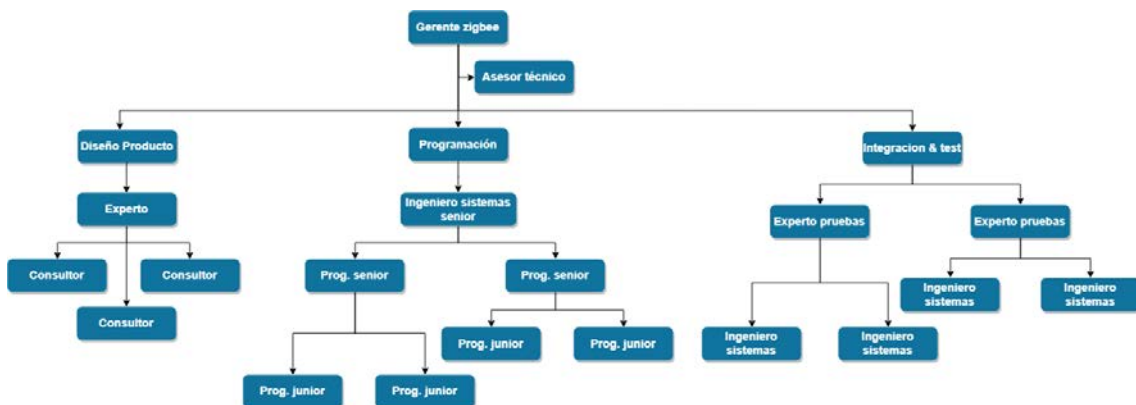


Ilustración 13: Organigrama software zigbee

Realizando cada uno de los cargos las siguientes funciones:

- **Gerente zigbee:** encargado de realizar los requerimientos y de coordinar el equipo de trabajo de esta parte del proyecto.
- **Asesor técnico:** dará soporte al gerente para cubrir los aspectos más técnicos del proyecto.
- **Experto:** responsable del diseño del software de zigbee.
- **Consultores:** encargados de realizar el diseño del software zigbee.
- **Ingeniero de sistemas senior:** responsable del desarrollo del software zigbee.
- **Programadores:** encargados de realizar el software zigbee.
- **Experto pruebas:** responsables de una parte del equipo de pruebas de integración y test.
- **Ingenieros de sistemas:** encargados de realizar las pruebas de integración y test.

11 PLANIFICACIÓN

Hemos obtenido toda la información necesaria para poder realizar una buena planificación, obteniendo todos los recursos que se necesitan al igual que el tiempo total que se necesita para realizar el proyecto.

Con estos datos hemos realizado una planificación que cumple con la información obtenida:

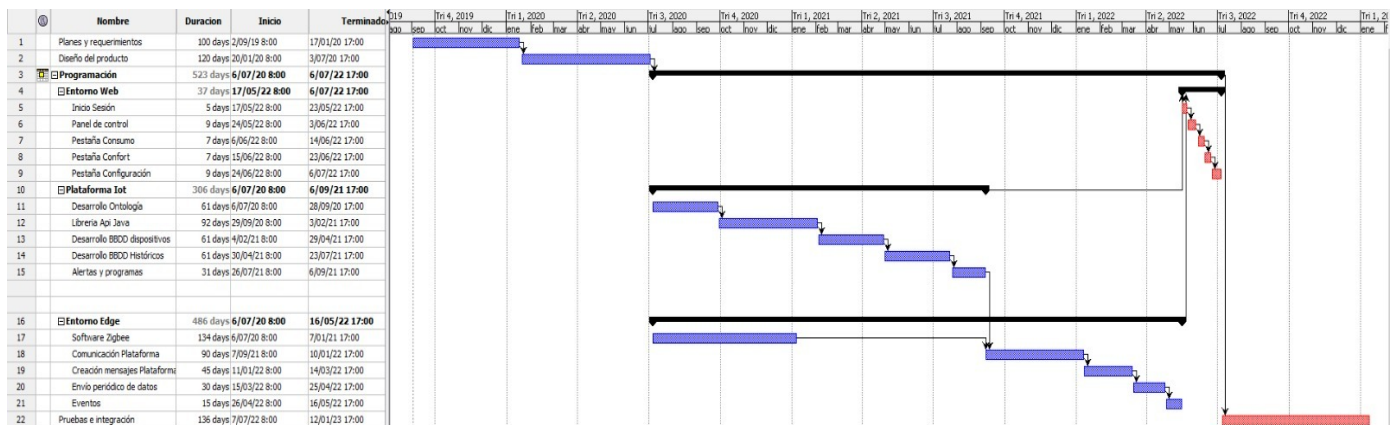


Ilustración 14: Cronograma planificación

El cronograma obtenido anteriormente se ha obtenido teniendo en cuenta las siguientes premisas.

1. El diseño de cada uno de los módulos no puede comenzar hasta que no se hallan visto todos los requerimientos del proyecto.
2. La programación no puede comenzar hasta que este está completado el diseño de cada uno de los módulos.
3. Antes de poder realizar el desarrollo de la web es necesario que el desarrollo de los demás módulos esté completo para poder obtener los datos necesarios y poder visualizarlos en la web.
4. El software de zigbee se puede comenzar cuando empieza la tarea de programación ya que se trata de un módulo aislado y no necesita interacción con otros módulos
5. El resto del desarrollo del entorno Edge tiene que realizarse una vez la plataforma Iot esté completa para poder implementar la comunicación con la plataforma.
6. Una vez se hayan completado todos los desarrollos de programación sería posible empezar con la integración y test.

1. Planificación y diseño

Antes de comenzar con el desarrollo del proyecto se ha realizado el análisis de requisitos que hay que cumplir para que todas las partes del proyecto se puedan desarrollar trabajando todos los equipos responsables de cada módulo y exponiendo cada uno sus necesidades.

Para obtener el tiempo que es necesario para realizar las tareas de planificación y diseño se ha optado por obtener el mayor tiempo de todos los módulos.

REQUERIMIENTOS	Entorno web	Plataforma Iot	Entorno edge	Software zigbee
Esfuerzo (PM)	2,9	24,59	1,2	10,79
Recursos	1	4	1	2
TOTAL	2,9	6,2	1,2	5,4

DISEÑO	Entorno web	Plataforma Iot	Entorno edge	Software zigbee
Esfuerzo (PM)	7,1	59,7	2,9	26,2
Recursos	1	8	1	4
TOTAL	7,1	7,5	2,9	6,5

Según los datos obtenidos, para los requerimientos se ha establecido que la duración será de 6,2 meses y el diseño será de 7,5 meses.

2. Programación y Pruebas

Para la duración del apartado de programación en cada uno de los módulos se ha establecido el tiempo prefijado por la aplicación de COCOMO II.

12 VISIÓN FUTURA

Hemos visto como realizar la estimación del software utilizando diferentes enfoques a la hora de realizarlo, pero lógicamente existe otras muchas formas de realizar la estimación que también podrían utilizarse para enforcarlo de otras formas, tales como utilizar puntos objetos en vez de puntos función o realizar la búsqueda de otros tipos de software en vez de utilizar COCOMO y COCOMO II.

El mundo de la estimación y planificación del software es muy amplio por lo que se podrían realizar el proyecto enfocándolo desde el punto de vista de la planificación con metodologías ágiles.

13 BIBLIOGRAFIA

1. Albatian.com. ¿Cómo guiar en la Madurez en BPM a las organizaciones? - Albatian. [online] Available at: <https://albatian.com/es/blog/como-guiar-en-la-madurez-en-bpm-a-las-organizaciones/>.
2. Blogadmi1.files.wordpress.com. [online] Available at: <https://blogadmi1.files.wordpress.com/2010/11/cocomo01full.pdf>.
3. Csse.usc.edu. COCOMO II - Constructive Cost Model. [online] Available at: <http://csse.usc.edu/tools/cocomoii.php>.
4. Csse.usc.edu. CSSE Website. [online] Available at: http://csse.usc.edu/csse/research/cocomoii/cocomo_downloads.htm.
5. Es.wikipedia.org. Gestión de proyectos. [online] Available at: https://es.wikipedia.org/wiki/Gesti%C3%B3n_de_proyectos.
6. Infor.uva.es. [online] Available at: <https://www.infor.uva.es/~manso/calidad/PFA-CLM-2011>.
7. Proyectos Ágiles. Qué es SCRUM. [online] Available at: <https://proyectosagiles.org/que-es-scrum/>.
8. Sc.ehu.es. El Modelo COCOMO. [online] Available at: <http://www.sc.ehu.es/jiwdocoj/mmis/cocomo.htm>.
9. Sc.ehu.es. MÉTODO DE ANÁLISIS PUNTOS DE FUNCIÓN MKII. [online] Available at: <http://www.sc.ehu.es/jiwdocoj/mmis/fpmkii.htm>.
10. Toptal Engineering Blog. Estimación de Costos de Software en Gestión de Proyectos Ágiles. [online] Available at: <https://www.toptal.com/agile/estimacion-de-costos-de-software-en-gestion-de-proyectos-agiles>.

14 ANEXOS

14.1 ANEXO I: MADUREZ DE UNA EMPRESA

BPMM (Business Process Maturity Model) establece que existen 5 niveles de madurez que una empresa, dependiendo de una serie de referentes a de cumplir para pertenecer a un nivel o a otro. Cada nivel tiene una serie de cosas puntuales que son más exigentes a medida que se va subiendo de nivel buscando una empresa madura, bien administrada y controlada.

Los procesos de madurez son los siguientes:

- **Nivel 1:** Los procesos suelen hacerse de forma específica y de una manera desordenada, no existe un entorno estable de trabajo por lo que el éxito de la empresa depende de la competencia y la disposición de los empleados que forman la empresa. Suele ocurrir que los productos que desarrollan funcionan, pero se suelen pasar en presupuesto y en calendario.
- **Nivel 2:** En este nivel se garantiza que los requisitos son debidamente gestionados y que los procesos se planifican, se miden, y se controlan correctamente. En este nivel los servicios y los trabajos con administrados.
- **Nivel 3:** En este nivel, aparte de cumplir con todos los requisitos del nivel 2 y 1, se establecen unos estándares y unas descripciones de los procesos y procedimientos por cada uno de los procesos que forman el proyecto, que pueden coincidir o no con el resto de procesos y procedimientos. Los procesos se describen con mucho detalle y de una forma rigurosa.
- **Nivel 4:** Todos los procesos que se realizan se controlan mediante técnicas estadísticas y otras técnicas cuantitativas. Si existen causas de variación de los procesos se identifican y se corrigen para que no sucedan de nuevo en el futuro.
- **Nivel 5:** En este nivel existe una mejora continua de los procesos a través de los aumentos y mejoras tecnológicas innovadoras.

14. 2 ANEXO II: TABLAS DE COMPLEJIDAD PUNTOS FUNCIÓN

En este apartado se muestran las tablas que se usan para clasificar la complejidad de cada una de las operaciones realizadas en el cálculo de los PF.

Esta tabla muestra la complejidad de cada EI o EQ basada en las cantidades de DET y RET:

EI	1-4 DET	5-15 DET	≥16 DET
0-1 FTR	Baja	Baja	Media
2 FTR	Baja	Media	Alta
≥ 3 FTR	Media	Alta	Alta

Siendo: Baja: 3; Media: 4; Alta: 6

Esta tabla muestra la complejidad de cada EO basada en las cantidades de DET y RET:

EO/EQ	1-5 DET	6-19 DET	≥20 DET
0-1 FTR	Baja	Baja	Media
2-3 FTR	Baja	Media	Alta
≥ 4 FTR	Media	Alta	Alta

Siendo: Baja: 4; Media: 5; Alta: 7

Esta tabla muestra la complejidad de cada ILF/EIF basada en las cantidades de DET y RET:

Complejidad	1 to 19 DET	20 to 50 DET	> 51 DET
1 RET	Baja	Baja	Media
2-5 RET	Baja	Media	Alta
> 6 RET	Media	Alta	Alta

Con los correspondientes valores para cada uno de los resultados.

Complejidad	Baja	Media	Alta
ILF	x7	x10	x15
EIF	x5	x7	x10

14.3 ANEXO III: TABLA PUNTOS DE FUNCIÓN AJUSTADOS

En esta tabla se encuentra la explicación de cada uno de los puntos que se deben aplicar para ajustar los PF obtenidos.

Cada uno de los puntos tiene un nivel de 0 a 5 dependiendo de lo específica que sea cada apartado.

CARACTERÍSTICA	DESCRIPCIÓN
Comunicación de datos	Cuántas facilidades de comunicación hay disponibles para ayudar en el intercambio de información con la aplicación o el sistema.
Procesamiento distribuido de datos	Cómo se manejan los datos y las funciones de procesamiento distribuido.
Rendimiento	Si existen requerimientos de velocidad o tiempo de respuesta.
Configuraciones fuertemente utilizadas	Cómo de intensivas se utilizan las plataformas hardware donde se ejecuta el sistema.
Frecuencia de transacciones	Con qué frecuencia se ejecutan las transacciones: Diariamente, semanalmente, ...
Entrada de datos on-line	Qué porcentaje de la información se ingresa on-line.
Eficiencia del usuario final	Aplicación diseñada para maximizar la eficiencia del usuario final.
Actualizaciones Online	Cuántos Archivos Lógicos Internos se actualizan por una transacción on-line.
Procesamiento complejo	Hay procesamientos lógicos o matemáticos intensivos en la aplicación.
Reusabilidad	La aplicación se desarrolla para suplir una o muchas de las necesidades de los usuarios.
Facilidad de instalación	Qué tan difícil es la instalación y la conversión al nuevo sistema.
Facilidad de operación	Cómo de efectivos y/o automatizados deben ser los procedimientos de arranque, parada y restauración
Instalación en distintos lugares	La aplicación fue concebida para su instalación en múltiples sitios y organizaciones.
Facilidad de cambio	La aplicación fue concebida para facilitar los cambios sobre la misma.

0 = No presente o sin influencia 3 = Influencia media
 1 = Influencia ocasional 4 = Influencia significativa
 2 = Influencia moderada 5 = Influencia muy fuerte

Comunicación de Datos (Valores)

- 0: Sistema aislado del exterior
- 1: Batch, usa periféricos E o S remotos
- 2: Batch, usa periféricos E y S remotos
- 3: Captura de datos en línea o teleproceso que pasa los datos o sistema de consulta
- 4: Varios teleprocesos con mismo protocolo
- 5: Varios protocolos. Sistema Abierto y con interfaces de todo tipo al exterior.

Proceso distribuido (Valores)

- 0: Sistema totalmente centralizado
- 1: Sistema realiza procesos en un equipo, salidas usadas vía software por otros equipos
- 2: Sistema captura, los trata en otro
- 3: Proceso distribuido, trans. una sola dirección
- 4: Ídem, transferencia en ambas direcciones.
- 5: procesos cooperantes ejecutándose en distintos equipos.

Objetivo de Rendimiento (Valores)

- 0: Rendimiento normal (no se da énfasis).
- 1: Se indican requisitos, no medida especial.
- 2: Crítico en algunos momentos. Procesos acabados antes de próxima sesión de trabajo.
- 3: Tiempo de respuesta es crítico.
- 4: ... en diseño hacer análisis de rendimiento en tiempo respuesta o cantidad operaciones/hora.
- 5: .. uso herramientas para alcanzar el rendimiento demandado por el usuario.

Integración de la aplicación (Valores)

- 0: No se indican restricciones
- 1: Existen las restricciones usuales
- 2: Características de seguridad o tiempos.
- 3: Restricciones en algún procesador
- 4: El software deberá funcionar con restricciones de uso en algún procesador.
- 5: Restricciones especiales para aplicación en los componentes distribuidos del sistema

Tasa de transacciones (Valores)

- 0: No se prevén picos.
- 1: Se prevén picos poco frecuentes (mensual).
- 2: Se prevén picos semanales.
- 3: Se prevén horas punta, diarias.
- 4: Tasa de trans. tan elevada que en diseño se hace análisis de rendimiento.
- 5: Análisis de rendimiento en diseño, implementación e instalación.

Entrada de datos on-line (Valores)

- 0: Todo es Batch.
- 1: 1% < entradas interactivas < 7%.
- 2: 8% < entradas interactivas < 15%.
- 3: 16% < entradas interactivas < 23%.
- 4: 24% < entradas interactivas < 30%.
- 5: Entradas interactivas > 30%.

Tipos de elementos asociados a la eficiencia del usuario

- Menús.
- Uso de ratón.
- Ayudas "en línea".
- Movimiento automático del cursor.
- Efectos de scroll (papiro).
- Teclas de función predefinidas.
- Lanzamiento de procesos Batch desde las transacciones "en línea".
- Selección mediante cursor de datos de la pantalla;
- Pantallas con muchos colores y efectos;
- Posibilidad de "hard-copy".
- Ventanas de "pop-up";
- Aplicación bilingüe (cuenta por cuatro).
- Aplicación Multilingüe (más de dos, cuenta por seis).

Valores

0: No se da énfasis al tema

1: 1 a 3 de los factores

2: 4 a 5 de los factores

3: 6 o más factores, sin requerir eficiencia

4: ... con requerimientos que implican estudio de los factores humanos en el diseño

5: ... se demandan prototipos y herramientas para verificar que se alcanzaran los objetivos.

Actualizaciones On-line (Valores)

0: No hay.

1: De 1 a 3 ficheros con información de control; cantidad baja y ficheros recuperables.

2: ... pero con 4 o más ficheros de control

3: Actualización de ficheros importantes

4: ... esencial la protección ante pérdidas

5: Gran cantidad de actualizaciones interactivas; sistemas de recuperación muy automatizados.

Lógica de Proceso Interno Compleja

La complejidad interna en un proceso está en función de las siguientes características:

- Especificados algoritmos matemáticos complejos.
- Proceso con lógica compleja.
- Especificado muchas excepciones, consecuencia de transacciones incompletas, que deberán tratarse.
- Manejar múltiples dispositivos de entrada / salida.
- Se incorporarán sistemas de seguridad y control.

Valores

0: Ninguna de las características.

1: 1 Característica.

2: 2 Características.

...

5: Las 5 características.

Reusabilidad del Código (Valores)

- 0: No se prevé.
- 1: Reutilizar código en la misma aplicación.
- 2: Menos de un 10% de la aplicación tiene en cuenta las necesidades de + de 1 usuario.
- 3: El 10 % o más ...
- 4: Aplicación preparada para ser reutilizable a nivel de código.
- 5: Aplicación preparada para ser reutilizable por medio de parámetros.

Contempla la conversión e instalación (Valores)

- 0: No se requiere conversión.
- 1: Se solicita facilidad de instalación.
- 2: Se solicitan procesos de conversión e instalación, no importantes para el proyecto.
- 3: Se solicitan procesos de conversión e instalación si son importantes para el proyecto
- 4: 2 y herramientas conversión e instalación.
- 5: 3 y herramientas conversión e instalación; sistema crítico para la empresa.

Facilidad de Operación (Valores)

- 0: No se especifica nada.
- 1 a 4: Sumar ítems según se cumplan: Procesos de arranque, back-up y recuperación con o sin intervención del operador y manejo de cintas u otros dispositivos manual o automático
- 5: Sistema automático sin intervención humana.

Instalaciones Múltiples (Valores)

- 0: 1 solo lugar.
- 1: Múltiples lugares, mismo hardware y software.
- 2: En diseño se tiene en cuenta el caso (1).
- 3: En diseño se tiene en cuenta múltiples entornos hardware y software.
- 4: Se documenta y planea para (1) y (2).
- 5: Ídem, para (3).

Facilidad de Cambios

Ítems a tener en cuenta:

- Consultas flexibles del usuario:
 - Simples con condiciones. lógicas and/or que implican un único fichero lógico.
 - Medias con condiciones lógicas sobre más de 1 F.L. (por 2).
 - Complejas con condiciones lógicas complejas que afectan a varios F.L. (por 3).
- Parámetros de la aplicación con tablas ajenas al código:
 - El cambio se hace efectivo al arrancar el sistema.
 - El cambio es interactivo (por 2).

Facilidad de Cambios (Valores)

0: No se especifica nada

1: Un ítem de valor 1

2: Ítems por valor 2

3: ...

5: Ítems por valor 5

14. 4 ANEXO IV: FACTORES DE AJUSTE (EM)

El significado de los atributos es el siguiente, según su tipo:

- Software

Atributo	Descripción
RELY	Garantía de funcionamiento requerida al software.
DATA	Tamaño de la base de datos en relación con el tamaño del programa.
CPLX	Representa la complejidad del producto.

- Hardware

Atributo	Descripción
TIME	Limitaciones en el porcentaje del uso de la CPU.
STOR	Limitaciones en el porcentaje del uso de la memoria.
VIRT	Volatilidad de la máquina virtual.
TURN	Tiempo de respuesta requerido.

- Personal

Atributo	Descripción
ACAP	Calificación de los analistas.
AEXP	Experiencia del personal en aplicaciones similares.
PCAP	Calificación de los programadores.
VEXP	Experiencia del personal en la máquina virtual.
LEXP	Experiencia en el lenguaje de programación a usar.

- Proyecto

Atributo	Descripción
MODP	Uso de prácticas modernas de programación.
TOOL	Uso de herramientas de desarrollo de software.
SCED	Limitaciones en el cumplimiento de la planificación.

A continuación, se muestra la valoración de cada atributo de acuerdo a su calificación, se muestra en la siguiente tabla

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
RELY	0,75	0,88	1	1,15	1,4	
DATA		0,94	1	1,08	1,16	
CPLX	0,7	0,85	1	1,15	1,3	1,65
Atributos de hardware						
TIME			1	1,11	1,3	1,66
STOR			1	1,06	1,21	1,56
VIRT		0,87	1	1,15	1,3	
TURN		0,87	1	1,07	1,15	
Atributos de personal						
ACAP	1,46	1,19	1	0,86	0,71	
AEXP	1,29	1,13	1	0,91	0,82	
PCAP	1,42	1,17	1	0,86	0,7	
VEXP	1,21	1,1	1	0,9		
LEXP	1,14	1,07	1	0,95		
Atributos del proyecto						
MODP	1,24	1,1	1	0,91	0,82	
TOOL	1,24	1,1	1	0,91	0,83	
SCED	1,22	1,08	1	1,04	1,1	

14. 5 ANEXO V: FORMULARIO SHEF

Proyecto: Sistema de Administración de Trabajo para Estudiantes													Analista: Juan Pérez			Fecha: 25/10/85									
1 Nro SS	02 Subsistema SS	8 SLOC	Producto		Atributos de la Computadora				Personal		Proyecto			32 EAF SS	20 PM MOD	33 PM EST	34 Costo Mes- Persona a	35 Costo	36 Totales Prom.						
			21 RELY	22 DATA	23 TIME	24 STOR	25 VIRT	26 TURN	27 ACAP	28 AEXP	29 MODP	30 TOOL	31 SCED												
1	Query	3700	PD 1.00 DD 1.00 CT 1.00 IT 1.00	1.00	1.00	1.00	1.00	0.95 0.90 0.85 0.80	1.00	0.75 0.90 0.90 0.85	0.75 0.80 0.85 0.85	1.00 0.95 0.90 0.83	0.98 0.95 0.90 0.85	1.00	0.52 0.58 0.53 0.41	2.0 3.2 5.5 2.5	1.0 1.9 2.9 1.0	6.0 5.0 5.0 5.5	6.0 9.5 14.5 5.5	6.8' 35.5' 544' 10'					
2	Update	2600	1.00	1.00	1.00	1.00	0.95 0.90 0.85 0.80	1.00	0.75 0.90 0.90 0.85	0.75 0.80 0.85 0.85	1.00 0.95 0.90 0.83	0.98 0.95 0.90 0.85	1.00	0.52 0.58 0.53 0.41	1.4 2.4 4.1 1.9	0.7 1.4 2.2 0.8	6.0 5.0 5.0 5.5	4.2 7.0 11.0 4.4	5.1 26.6 510 10						
3	Utilities	1700	1.00	1.00	1.00	1.00	0.95 0.90 0.85 0.80	1.00	1.00	1.00	1.00 0.95 0.90 0.83	0.98 0.95 0.90 0.85	1.00	0.93 0.81 0.69 0.56	0.7 1.4 2.5 1.2	0.7 1.1 1.7 0.7	5.0 5.0 5.5 5.5	3.5 5.5 9.4 3.8	4.2 22.2 405 13						
																			Product. Estimada						
Modo de Desarrollo Orgánico		9	8.000	Total SLOC														38	2.4 4.4 6.8 2.5			13.7 22.0 34.9 13.7	SLOC/PM 8000/16.1= 496.8—497		
10			28.4	Esfuerzo Nominal PM _{Nominal}														39	Total		16.1	7.2	84.3	Costo Total p/Instruc. 84.3/8—11	
11			282	Productividad Nominal (SLOC/PM) _{Nominal}														PM _{Estimado}		Cronograma TDEV		Costo Total del Sistema			
																		12		% de Distribución del PM por Fase		16 25 40 19		1 Esfuerzo Estimado por Subsistema 2 Costo Total del Subsistema en \$ 3 Productividad 4 Costo por instrucción	