

Estimación del proyecto Software

v3.0.1

Ingeniería del Software



PiKey Team- P_KT :

Jesús Aguirre Pemán
Enrique Ballesteros Horcajo
Jaime Dan Porras Rhee
Ignacio Iker Prado Rujas
Alejandro Villarín Prieto

3 de Abril de 2013

Índice general

Resumen

Este documento se puede considerar como una ampliación de una parte del **Plan de proyecto** del desarrollo de una aplicación para el campo de la hostelería. Más concretamente, vamos a estimar esfuerzo, coste y duración mediante la técnica de Puntos de Función (también conocida como FPA o Function Point Analysis), un procedimiento de descomposición basado en el problema. Esta métrica fue introducida en 1979 por Allan Albrecht, y existen distintas metodologías de medición. La más conocida es la del IF-PUG (International Function Points Users Group), que es una organización que ofrece certificación a especialistas de puntos de función de todo el mundo.

En cuanto al documento en sí, comenzaremos dividiendo el producto en partes más pequeñas para luego centrarnos en cada una de ellas, pues de eso trata una técnica basada en descomposición. Éstas partes, de forma general, son clientes, empleados, restaurante, hotel. De cada una de estas fracciones obtendremos, tras un análisis, un determinado número de puntos de función, obedeciendo al número de entradas-salidas, los ficheros accedidos y las interfaces, así como la dificultad de los mismos.

Una vez vez hecho esto, calcularemos el factor de ajuste para obtener los puntos de función ajustados. Entonces estaremos en condiciones de estimar y obtener unos resultados que se podrán aplicar al proyecto. Para ello utilizaremos la herramienta COCOMO II.2000.4, que se puede encontrar en

http://csse.usc.edu/csse/research/COCOMOII/cocomo_downloads.htm

Parte I

Técnicas de estimación

1. Determinación del tipo de cuenta de PF

Al tratarse del inicio del proyecto, estamos trabajando con una estimación de los puntos de función, que diferirá de los puntos de función de la aplicación ya finalizada, dependiendo de los cambios que se vayan produciendo.

Por otro lado, estamos trabajando en un proyecto de desarrollo de un programa para la gestión de un negocio hostelero, no se trata del mantenimiento o mejora de otro ya existente.

2. Determinación de los límites del sistema

En esta sección se concreta el límite entre la aplicación de hostelería y otros sistemas que serán necesarios para su uso adecuado.

El sistema tiene una serie de funcionalidades, que se pueden revisar en el **Documento de especificación de requisitos Software**, pero no todas nacen de la propia aplicación. Las facturas y los C.V. se abren en PDF, utilizando el visor de PDF que tenga el equipo sobre el que estemos. También permite, al generar las facturas, exportarlas a PDF para enviárselas por correo electrónico a los clientes si así lo solicitan. La modificación de información, se actualizará rápidamente para un mejor funcionamiento del negocio. Cuanto más veloz es, más rápidamente se producen las acciones y por tanto más eficiente puede ser el negocio. Por otro lado, en la aplicación o en los PDFs pueden aparecer hipervínculos a una dirección de Internet (página web de la lavandería, del hotel...), que si se seleccionan lanzarían el navegador por defecto del equipo.

En cuanto a la gestión de toda esta información, los datos se recogerán en ficheros auxiliares, protegidos en un servidor interno. Principalmente podemos diferenciar tres partes: lo relacionado con empleados, con clientes y datos de la aplicación (como el menú, las existencias, manteles en la lavandería, preferencias del sistema...). Los ficheros estarán ocultos, y sólo la aplicación podrá acceder a ellos. En principio, el sistema trabajará directamente sobre los mismos, salvándolos al menos una vez por minuto (protegiéndose así de los apagones o el agotamiento de batería).

3. Cálculo de los PF sin ajustar

3.1. Entradas Externas

Las Entradas Externas o EI (External Inputs) son procesos que hacen llegar información desde el exterior a nuestra aplicación, ya sea desde un usuario o desde otra aplicación. Siempre que tengamos una entrada externa, se modificará un ILF (luego pasaremos a estudiarlos). A continuación se estudia cuántas EI hay y qué complejidad tienen.

La complejidad viene dada por dos conceptos: DET (Data Element Type), que es un campo individual identificable por el usuario y modifica un ILF, y FTR (File Type Referenced), que es el ILF o EIF que interviene. Entonces la complejidad queda:

FTR\DET	De 1 a 4 DET	De 5 a 15 DET	16 o más DET
0 o 1 FTR	Baja	Baja	Media
2 FTR	Baja	Media	Alta
3 o más FTR	Media	Alta	Alta

Pasamos a detallar las EI:

■ Log In:

- DET: Dos campos (Usuario y contraseña) en el caso de acceder mediante teclado, y un sólo campo si se accede mediante lectura de un código de barras (o similares) de una tarjeta del empleado. Contamos dos.
- FTR: *Log In* es el único fichero al que se accede, donde se encuentra la información de los empleados, que permite acceder al sistema.
- Complejidad: Baja.

■ Cuenta de caja:

- DET: Tres atributos, caja de recepción o de restaurante, ingreso o gasto y concepto.
- FTR: *Cuenta de caja de recepción* y *Cuenta de caja de restaurante* son los ficheros que se modifican, pero cada vez que añades algo sólo modificas uno.
- Complejidad: Baja.

■ Libro diario:

- DET: Cuatro campos: cantidad de la cuenta del debe, concepto del debe, cantidad de la cuenta del haber y concepto del haber.
- FTR: *Libro diario* sólo.
- Complejidad: Baja.

■ Libro mayor:

- DET: Cuatro campos: fecha, contrapartida, importe y saldo total del ejercicio.
- FTR: *Libro mayor*.
- Complejidad: Baja.

■ Empleados:

- DET: Para dar de alta a un nuevo empleado hacen falta doce campos, y son nombre completo, DNI, sexo, edad, domicilio, nacionalidad, estado civil, número de teléfono, salario bruto, fotografía de carnet, C.V. y comentarios.
- FTR: Interviene el fichero *Empleados* y *Curriculums*.
- Complejidad: Media.

- Clientes:
 - DET: Para añadir un nuevo cliente, son necesarios siete atributos: nombre completo, DNI, número de teléfono, V.I.P., facturas, número de visitas en los últimos 12 meses, comentarios.
 - FTR: Fichero *Clientes*.
 - Complejidad: Baja.
- Reservas para el restaurante:
 - DET: Para hacer una reserva en el restaurante, cuatro campos: fecha, hora, nombre y número de comensales.
 - FTR: *Reservas restaurante*.
 - Complejidad: Baja.
- Pedidos:
 - DET: Tres campos, a saber, número de mesa, elemento solicitado y cantidad.
 - FTR: *Pedidos*.
 - Complejidad: Baja.
- Generar factura del restaurante:
 - DET: Número de mesa.
 - FTR: Se accede a *Pedidos* y *Clientes* para adjuntar la factura al cliente correspondiente.
 - Complejidad: Baja.
- Existencias:
 - DET: Para llevar el control de existencias, se puede guardar qué se necesita comprar y en qué cantidad: dos campos.
 - FTR: *Existencias*.
 - Complejidad: Baja.
- Reservas para el hotel:
 - DET: Siete campos, nombre, fecha de inicio, fecha de salida, número de inquilinos, número de niños, tipo de pensión y número de habitación.
 - FTR: El fichero *Reservas habitaciones*.
 - Complejidad: Baja.
- Generar factura del hotel:
 - DET: Número de habitación.

- FTR: Se accede a *Reservas habitaciones* y a *Clientes*.
- Complejidad: Baja.
- Añadir notas:
 - DET: El cuerpo de la nota. Recordamos que las notas son una vía de comunicación entre todos los trabajadores del negocio.
 - FTR: Un fichero con las notas (*Notas*).
 - Complejidad: Baja.
- Notificar incidencias:
 - DET: Dos campos: lugar y descripción.
 - FTR: *Incidencias*.
 - Complejidad: Baja.
- Determinar tareas a realizar (limpieza):
 - DET: Cuatro atributos, que son: empleado, fecha, hora y descripción de la tarea.
 - FTR: El fichero *Tareas limpieza* solamente.
 - Complejidad: Baja.
- Notificar tareas realizadas (limpieza):
 - DET: Un campo, la tarea que se ha realizado.
 - FTR: *Tareas limpieza* es el único fichero involucrado.
 - Complejidad: Baja.
- Lavandería:
 - DET: Habitualmente, la limpieza de sábanas, toallas, servilletas y manteles no se realiza en el propio hotel-restaurant, si no que se subcontrata a una empresa de lavandería para que lo haga. Esta funcionalidad permite llevar un recuento de los envíos que se han realizado. Fecha de salida a lavandería, número de manteles, número de servilletas, número de paquetes de sábanas y número de toallas, lo cual son cinco campos.
 - FTR: Fichero *Lavandería*.
 - Complejidad: Baja.

Por tanto podemos clasificar las EI en función de su complejidad usando la tabla antes presentada:

- **BAJA: 16 entradas externas.**

- **MEDIA: 1 entradas externas.**
- **ALTA: Ninguna.**

La forma en que hemos procedido es la siguiente en todos los casos: Primero, fijándonos en el número de campos, identificamos en qué columna estamos de la tabla, y después, con el número de FTRs la fila. Por ejemplo para el primer caso, el Log In, vimos que había dos campos (usuario y contraseña), por lo que con dos DET estamos en la segunda columna de la tabla (**De 1 a 4 DET**) y un sólo fichero *Log In*, que nos sitúa en la segunda columna (**0 o 1 FTR**). Por tanto la entrada del Log In tiene una complejidad baja.

3.2. Salidas Externas

Las Salidas Externas o EO (External Outputs) son procesos que hacen llegar información desde la aplicación al exterior, ya sea a un usuario o a otra aplicación. A continuación se estudia cuántas EO hay y qué complejidad tienen.

La complejidad viene dada por dos conceptos: DET (Data Element Type), que es un campo individual identificable por el usuario y aparece en la EO, y FTR (File Type Referenced), que es el ILF o EIF que interviene. La complejidad es bastante parecida a las EI:

FTR\DET	De 1 a 5 DET	De 6 a 19 DET	20 o más DET
0 o 1 FTR	Baja	Baja	Media
2 o 3 FTR	Baja	Media	Alta
4 o más FTR	Media	Alta	Alta

Pasamos a detallar las EO:

- Cuenta de caja:
 - DET: Muestra tres atributos, caja de recepción o de restaurante, ingreso o gasto y concepto.
 - FTR: *Cuenta de caja de recepción* y *Cuenta de caja de restaurante* son los ficheros de donde se lee.
 - Complejidad: Baja.
- Libro diario:
 - DET: Cuatro campos: la cantidad de la cuenta del debe, el concepto del debe, la cantidad de la cuenta del haber y el concepto del haber.
 - FTR: *Libro diario* sólo.
 - Complejidad: Baja.
- Libro mayor:

- DET: Cuatro campos: la fecha, la contrapartida, el importe y el saldo total del ejercicio, mostrados por columnas.
- FTR: *Libro mayor*.
- Complejidad: Baja.
- Empleados:
 - DET: Al mostrar la información de un empleado hay dos formas, en modo lista, donde sólo se muestra nombre completo, foto, y si trabaja actualmente en la empresa, o la versión detallada, con doce campos, los cuales son nombre completo, DNI, sexo, edad, domicilio, nacionalidad, estado civil, número de teléfono, salario bruto, fotografía de carnet, C.V. y comentarios. Como son independientes (pues en principio el usuario estará mirando en modo lista y luego clicará uno para verlo en detalle) cuentan como dos salidas distintas.
 - FTR: Interviene el fichero *Empleados y Curriculum*s.
 - Complejidad: El primero (vista sencilla) baja, y el segundo (detallada) Media.
- Clientes:
 - DET: De igual manera, aquí también tenemos dos salidas independientes. En la vista de lista aparecerá nombre completo, número telefónico y si es o no V.I.P., mientras que en la vista detallada los atributos son: nombre completo, DNI, número de teléfono, V.I.P., facturas, número de visitas en los últimos 12 meses, comentarios.
 - FTR: Fichero *Clientes*, que contiene toda la información de los mismos.
 - Complejidad: Ambos baja.
- Ver las reservas del restaurante:
 - DET: En las reservas del restaurante aparecen cinco campos: día, hora, nombre, número de comensales y número de mesa.
 - FTR: *Reservas restaurante*.
 - Complejidad: Baja.
- Ver la distribución de las mesas:
 - DET: Requiere de muchos atributos, además de la forma de la sala donde está el restaurante, para cada mesa el tamaño y la posición.
 - FTR: *Dimensiones restaurante, Mesas del restaurante y Posición de las mesas*.
 - Complejidad: Alta, teniendo en cuenta el número de mesas y los ficheros a los que accede.
- Ver un pedido:

- DET: Varía en función del momento en el que se consulte y de la mesa, pero teniendo en cuenta bebida, comida, precio y cantidad de cada uno de esos items habrá más de 20 (si simplemente se tienen tres bebidas distintas y cuatro entrantes ya hay 21 atributos).
- FTR: *Pedidos*.
- Complejidad: Media.
- Ver el menú:
 - DET: Debido a la extensión de un Menú habitual, se superarán las 20.
 - FTR: Se accede a *Menú* y *Existencias*.
 - Complejidad: Alta.
- Ver las existencias:
 - DET: Por el mismo motivo que en el caso anterior, se maneja mucha información.
 - FTR: *Existencias*.
 - Complejidad: Media.
- Ver las reservas del hotel:
 - DET: Se muestran siete campos en modo lista, nombre, fecha de inicio, fecha de salida, número de inquilinos, número de niños, tipo de pensión y número de habitación, con sólo tres reservas ya se superan los 20 campos.
 - FTR: El fichero *Reservas habitaciones*.
 - Complejidad: Alta.
- Ver un habitación:
 - DET: Permite conocer para una cierta habitación la planta en la que se encuentra, si se accede mediante ascensor o escaleras, el tipo de ducha (bañera/jacuzzi), y una breve descripción de la misma.
 - FTR: Se accede a *Habitaciones*.
 - Complejidad: Media, debido al número de habitaciones y a que sólo se accede a un fichero.
- Ver las notas:
 - DET: El cuerpo de la nota de varias notas (al menos el comienzo).
 - FTR: Un fichero con las notas (*Notas*).
 - Complejidad: Media si se utiliza mínimamente este servicio, y se supone que así será pues para eso lo encarga el cliente.

- Ver las incidencias:
 - DET: Para cada incidencia, dos campos: lugar y descripción.
 - FTR: *Incidencias*.
 - Complejidad: Baja en principio, a no ser que se tengan continuas averías y por tanto haya más de diez incidencias a la vez.
- Ver las tareas a realizar (limpieza):
 - DET: Tres atributos, la fecha, la hora y la descripción de la tarea por realizar.
 - FTR: *Tareas limpieza* es el único fichero involucrado.
 - Complejidad: Baja.

Por tanto podemos clasificar las EO en función de su complejidad usando la tabla antes presentada:

- **BAJA: 9 salidas externas.**
- **MEDIA: 5 salidas externas.**
- **ALTA: 3 salidas externas.**

El procedimiento para la obtención de la complejidad ha sido exactamente el mismo que antes, pero con la tabla correspondiente.

3.3. Consultas Externas

Las Consultas Externas o EQ (External Queries) son procesos que combinan la entrada y salida de información, dando lugar a una consulta a los datos. Cuando se produce una EQ no se modifican los datos del sistema. A continuación se estudia cuántas EQ hay y qué complejidad tienen.

La complejidad viene dada por dos conceptos, como siempre: DET (Data Element Type), que es un campo individual identificable por el usuario de entrada o salida, y FTR (File Type Referenced), que es el ILF o EIF que interviene. Es importante notar que ahora la complejidad no se obtiene como una suma, si no como el máximo entre la consulta de entrada y la de salida. La complejidad queda como en las EI:

FTR\DET	De 1 a 4 DET	De 5 a 15 DET	16 o más DET
0 o 1 FTR	Baja	Baja	Media
2 FTR	Baja	Media	Alta
3 o más FTR	Media	Alta	Alta

Pasamos a detallar las EQ:

- Editar un empleado:

- DET: Como al editar los datos de un empleado el usuario como mucho va a modificar todos los campos, la información de salida será siempre mayor o igual que los elementos de entrada. Como antes hemos visto que hay doce campos para un empleado, ese es el número de atributos.
- FTR: *Empleados* y *Curriculums* son los ficheros a los que se accede.
- Complejidad: Media, por tener 12 atributos y dos FTRs.
- Editar un cliente:
 - DET: Razonando igual, obtenemos siete atributos.
 - FTR: *Cientes* sólo.
 - Complejidad: Baja.
- Exportar a PDF:
 - DET: Se podrán exportar a PDF datos del libro mayor o diario, facturas... Hay más datos de salida, que trabajan con el programa de conversión a PDFs correspondiente.
 - FTR: En función de lo que se exporte se tratará de un fichero u otro, pero sólo uno.
 - Complejidad: Media.
- Ver y cuantificar existencias:
 - DET: Mientras navegas entre las existencias, puedes marcar las que sean escasas. Por supuesto, la información de salida es mucho mayor que la de entrada, pues muestra todas las existencias, y cada una con dos campos: concepto y cantidad.
 - FTR: El fichero de *Existencias*.
 - Complejidad: Media.
- Editar la distribución de las mesas:
 - DET: Aquí aparecen muchos atributos, además de la forma de la sala donde está el restaurante, para cada mesa el tamaño y la posición. Por supuesto, tomamos los datos de salida por ser mayores o iguales en número que los de entrada.
 - FTR: Los ficheros son *Dimensiones restaurante*, *Mesas del restaurante* y *Posición de las mesas*.
 - Complejidad: Alta, pues maneja tres ficheros y mucha información.
- Editar una habitación:

- DET: Se pueden modificar campos de la descripción de las habitaciones, lo cual puede pasar por ejemplo tras una reforma, o una renovación del mobiliario. El número de entradas es menor o igual que el de salidas, que es cuatro (visto en la sección de EO).
 - FTR: El fichero *Habitaciones*.
 - Complejidad: Baja.
- Editar una nota:
- DET: Las notas se pueden modificar, o eliminar permanentemente. Hay un único campo, sea de entrada o salida, con el cuerpo de la nota.
 - FTR: El fichero con las *Notas*.
 - Complejidad: Baja.

Por ello, podemos clasificar las EQ en función de su complejidad usando la tabla antes presentada:

- **BAJA: 3 consultas externas.**
- **MEDIA: 3 consultas externas.**
- **ALTA: 1 consulta externa.**

3.4. Ficheros Lógicos Internos

Los Ficheros Lógicos Internos o ILF (Internal Logical Files) son un conjunto de datos que el usuario puede identificar, que están relacionados entre sí, y que son mantenidos por la aplicación. A continuación se estudia cuántos ILF hay y qué complejidad tienen.

La complejidad viene dada por dos conceptos: DET (Data Element Type), que es un campo individual identificable por el usuario, y RET (Record Element Type), que es un subgrupo de datos. Entonces la complejidad queda:

RET\DET	De 1 a 19 DET	De 20 a 50 DET	51 o más DET
1 RET	Baja	Baja	Media
De 2 a 5 RET	Baja	Media	Alta
6 o más RET	Media	Alta	Alta

Pasamos a detallar los ILF:

- Log In:
 - DET: Tres campos en la tabla, como mucho: usuario, contraseña y código para el acceso mediante tarjeta.
 - RET: Una única tabla en el *Log In*.
 - Complejidad: Baja.

- Cuenta de caja (restaurante y recepción):
 - DET: Tres campos, tipo de caja, cantidad del ingreso o gasto y concepto.
 - RET: Dos ficheros: *Cuenta de caja de restaurante* y *Cuenta de caja de recepción*. Por tanto dos tablas.
 - Complejidad: Baja.
- Libros de contabilidad (diario y mayor):
 - DET: Cuatro campos en cada uno (descritos anteriormente).
 - RET: Dos tablas que se corresponden con *Libro diario* y *Libro mayor*.
 - Complejidad: Baja.
- Empleados:
 - DET: Dos tablas, una con los datos del empleado que se muestran (doce campos), y otra con los C.V.
 - RET: Ficheros que intervienen: *Empleados* y *Curriculums*, con una tabla en cada uno.
 - Complejidad: Baja.
- Clientes:
 - DET: Anteriormente vimos que son siete los atributos.
 - RET: *Clientes* es el único fichero, con una tabla para la información del cliente y otra con las facturas de cada cliente.
 - Complejidad: Baja.
- Reservas del restaurante:
 - DET: Cuatro campos.
 - RET: Una tabla de *Reservas restaurante*.
 - Complejidad: Baja.
- Pedidos del restaurante:
 - DET: Se puede considerar que los campos son todos los que ofrece el menú, y para cada uno de ellos la cantidad. Además de esto, el número de mesa.
 - RET: Una tabla en *Pedidos*.
 - Complejidad: Si la carta es muy extensa puede llegar a ser de complejidad media, pero vamos a suponer baja, ya que más de 51 DETs en el menú es mucho.
- Existencias:

- DET: Dos campos, elemento a comprar y cantidad.
- RET: En *Existencias* una única tabla.
- Complejidad: Baja.
- Reservas del hotel:
 - DET: Siete campos, analizados antes.
 - RET: En *Reservas habitaciones* sólo una tabla.
 - Complejidad: Baja.
- Notas:
 - DET: Sólo el cuerpo de la nota.
 - RET: Una tabla con las *Notas*.
 - Complejidad: Baja.
- Incidencias:
 - DET: Atributos de lugar y descripción.
 - RET: Una tabla en *Incidencias*.
 - Complejidad: Baja.
- Tareas a realizar (limpieza):
 - DET: Cuatro campos.
 - RET: Sólo una tabla, que se corresponde con el fichero de *Tareas de limpieza*.
 - Complejidad: Baja.
- Envíos a lavandería:
 - DET: Cinco campos, estudiados con anterioridad.
 - RET: Una tabla con los envíos, relativa a *Lavandería*.
 - Complejidad: Baja.

Tras este análisis, podemos clasificar los ILF en función de su complejidad usando la tabla antes presentada:

- **BAJA: 13 ficheros lógicos internos.**
- **MEDIA: Ninguno.**
- **ALTA: Ninguno.**

3.5. Ficheros de Interfaz Externos

Los Ficheros de Interfaz Externos o EIF (External Interface Files) son un conjunto de datos que el usuario puede identificar y que la aplicación referencia, pero son mantenidos por otra aplicación. Son los ILF pertenecientes a otra aplicación. Vamos a estudiar cuántos EIF hay y qué complejidad tienen.

La complejidad viene dada de la misma forma que en los ILF, y se resume en esta tabla:

RET\DET	De 1 a 19 DET	De 20 a 50 DET	51 o más DET
1 RET	Baja	Baja	Media
De 2 a 5 RET	Baja	Media	Alta
6 o más RET	Media	Alta	Alta

Apenas hay EIF en la aplicación, pues prácticamente toda la información se introduce manualmente o proviene de los ILFs del propio sistema. Aún así podemos distinguir tres casos:

- Abrir un PDF:
 - DET: Nombre del documento que se desea abrir.
 - RET: Una sola tabla.
 - Complejidad: Baja.
- Exportar a PDF:
 - DET: Texto que contendrá el PDF.
 - RET: Una única tabla con dicha información. El fichero accedido dependerá de lo que se exporte (por ejemplo para exportar datos del libro diario se accedería a *Libro diario*).
 - Complejidad: Baja.
- Lanzar el explorador de Internet:
 - DET: Solamente la dirección web a la que se quiere acceder.
 - RET: Una tabla.
 - Complejidad: Baja.

Así, podemos clasificar estos tres EIF utilizando su tabla:

- **BAJA: 3 ficheros de interfaz externos.**
- **MEDIA: Ninguno.**
- **ALTA: Ninguno.**

3.6. Cálculo efectivo de los PF sin ajustar

Para el cálculo de los puntos de función sin ajustar, utilizamos la información que acabamos de obtener. Sabemos ya, para cada tipo, la cantidad de elementos de cada complejidad, y para cada uno de éstos se define un peso. Utilizando estos datos, los PF sin ajustar se pueden resumir en la siguiente tabla:

Puntos de función	Complejidad						Total
	Simple		Media		Alta		
	#	Peso	#	Peso	#	Peso	
Entradas Externas	16	3	1	4	0	6	52
Salidas Internas	9	4	5	5	3	7	82
Consultas Externas	3	3	3	4	1	6	27
Ficheros Logicos Internos	13	7	0	10	0	15	91
Ficheros de Interfaz externos	3	5	0	7	0	10	15
Total de puntos de función sin ajustar:							267

Por tanto tenemos un total de 267 puntos de función sin ajustar. Ahora procedemos a calcular el factor de ajuste.

4. Cálculo del factor de ajuste

Existen catorce factores de complejidad, y a cada uno de ellos se le asocia un grado de influencia, entre cero y cinco:

- **0:** Sin influencia.
- **1:** Influencia mínima.
- **2:** Influencia moderada.
- **3:** Influencia moderada.
- **4:** Influencia significativa.
- **5:** Influencia apreciable.

1 Comunicación de datos: 2, ya que utiliza entrada y salida con impresión remotos (por ejemplo para que los cocineros vean lo que tienen que hacer).

2 Proceso distribuido: 0, ya que no existen.

3 Objetivos de rendimiento: 1, porque no estamos con un sistema crítico, pero hay requisitos de rendimiento para soportar varias operaciones con un mínimo de eficiencia.

4 Integración de la aplicación: 2, pues debe de responder en la tablet o equipo correspondiente aunque haya más aplicaciones abiertas. Además tiene restricciones de seguridad, no todos los empleados pueden ver toda la información del sistema.

5 Tasa de transacciones: 3. Habrá horas punta (como la comida), en las que haya más transacciones de información.

6 Entrada de datos interactiva: 5, ya que más del 30 % de las entradas son interactivas.

7 Eficiencia para el usuario final: 3. Hay más de seis factores, selección de datos de la pantalla, pantallas con muchos campos y colores, ventanas pop-up, efecto scroll, uso de ratón (en ordenadores), menús...

8 Actualizaciones interactivas: 5. Como hemos dicho, al menos una vez por minuto se deben guardar las modificaciones que el usuario haga para evitar la pérdida de información por apagones o agotamiento de batería.

9 Lógica de proceso interna compleja: 1, ya que no lleva una lógica muy complicada, pero hay sistema de seguridad.

10 Reusabilidad del código: 4. En principio, con pocas modificaciones de código se podría adaptar para otra empresa hostelera.

11 Conversión e instalación: 1, ya que no se definen requisitos especiales para la instalación, pero tiene que cumplir unas características para que sea sencillo.

12 Facilidad de operación: 4. Se presta gran atención al diseño, convirtiéndola en una aplicación vistosa pero sencilla, fácil de aprender a manejar. En posteriores documentos se profundizará en esto.

13 Instalaciones múltiples: 4, ya que se planea al menos tres plataformas (iOS, Android y Windows) con distinto hardware (tablets, smartphones computadora) y también se documenta.

14 Facilidad de cambios: 4. Consultas flexibles del usuario: medias (condiciones lógicas sobre más de un fichero lógico), y el cambio de parámetros de la aplicación con tablas ajenas al código es interactivo.

En la siguiente tabla se resume el cálculo del factor de complejidad total:

Id.	Factor de complejidad	Grado de influencia
1	Comunicación de datos	2
2	Proceso distribuido	0
3	Objetivos de rendimiento	1
4	Integración de la aplicación	2
5	Tasa de transacciones	3
6	Entrada de datos interactiva	5
7	Eficiencia para el usuario final	3
8	Actualizaciones interactivas	5
9	Lógica de proceso interna compleja	1
10	Reusabilidad del código	4
11	Conversión e instalación	1
12	Facilidad de operación	4
13	Instalaciones múltiples	4
14	Facilidad de cambios	4

Por tanto el factor de complejidad total o FCT es 39.

Finalmente, estamos en condiciones de calcular el factor de ajuste o AF:

$$AF = FCT \cdot 0,01 + 0,65 \Rightarrow AF = 39 \cdot 0,01 + 0,65 = 1,04$$

5. Cálculo de los PF ajustados

Al fin, estamos en condiciones de calcular los puntos de función ajustados (PFA)

$$PFA = PFSA \cdot AF = 267 \cdot 1,04 = 277,7$$

, siendo PFSA los puntos de función sin ajustar.

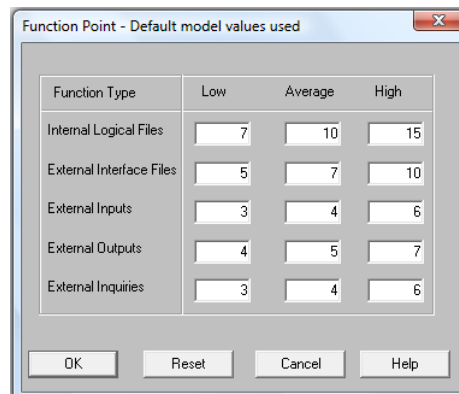
Parte II

Estimaciones de esfuerzo, coste y duración

1. Iniciando la herramienta

Para esta parte, como se dijo en el *Extracto*, se ha utilizado la herramienta COCOMO II.2000.4.

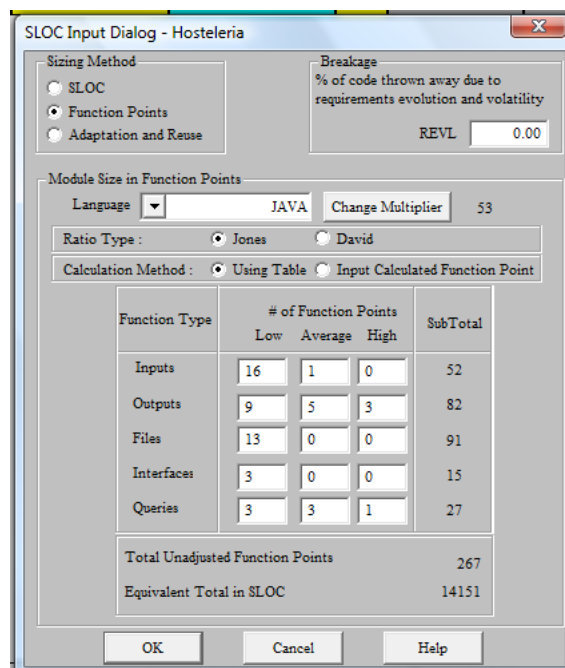
En primer lugar, se introdujo un nuevo módulo: *Hostelería*. Posteriormente, en *Module size* se añadió los datos sobre PF. El peso de cada uno es el habitual, seleccionado así:



Function Type	Low	Average	High
Internal Logical Files	7	10	15
External Interface Files	5	7	10
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6

Buttons: OK, Reset, Cancel, Help

El número de entradas, salidas etc. exactamente igual que antes:



Sizing Method: ☒ SLOC ☒ Function Points ☐ Adaptation and Reuse

Breakage: % of code thrown away due to requirements evolution and volatility. REVL: 0.00

Module Size in Function Points

Language: JAVA Change Multiplier: 53

Ratio Type: ☒ Jones ☐ David

Calculation Method: ☒ Using Table ☐ Input Calculated Function Point

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Inputs	16	1	0	52
Outputs	9	5	3	82
Files	13	0	0	91
Interfaces	3	0	0	15
Queries	3	3	1	27
Total Unadjusted Function Points				267
Equivalent Total in SLOC				14151

Buttons: OK, Cancel, Help

Con ello obtenemos una medida del tamaño de la aplicación. Nos da 14151 líneas de código.

2. Factores de escala (Scale Factor)

- PREC: Precedentness

El equipo de desarrolladores está compuesto por principiantes poco acostumbrados a trabajar con sistemas software, por tanto consideramos que este campo debe ser *muy bajo*.

- FLEX: Development Flexibility

La flexibilidad de desarrollo se fija como *baja* ya que hay unos requisitos predefinidos y que estamos trabajando con el Proceso Unificado.

- RESL: Architecture/Risk Resolution

El estudio de los riesgos se puede examinar en el documento de **Gestión de riesgos Software**, donde se ve que tenemos 5 riesgos con un nivel intolerable. Debido a esto y a la arquitectura, fijamos este valor como *bajo*.

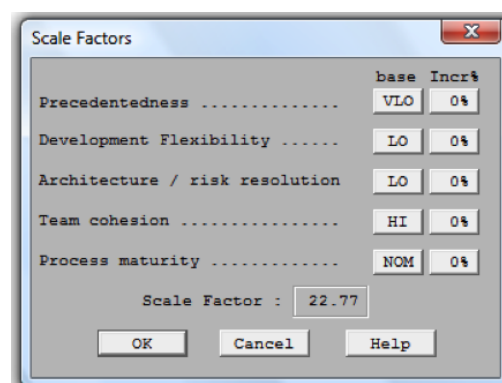
- TEAM: Team Cohesion

Ya que los miembros del equipo se conocen bien, hay buen ambiente de trabajo y se ven prácticamente todos los días, consideramos que la cohesión del equipo es *alta*. Además, se tiene bastante capacidad de acomodación a los intereses de otros stakeholders.

- PMAT: Process Maturity

Mirando el CMM Maturity Model hemos concluido que el proyecto se encuentra en un nivel 2, que es una madurez media. Por lo tanto queda como *nominal*.

Con estos datos, la herramienta determina que el factor de escala es 22.77, como se ve en la imagen.



3. Factor de ajuste del esfuerzo (EAF)

- Factores del producto

- RELY: Required Software Reliability

- El software debe tener una fiabilidad *alta*, sobre todo si se trata de una empresa mediana- grande.

- DATA: Data Base Size

- Ya que se manejan bases de datos de clientes, empleados y platos, y que de nuevo cuanto más grande sea el negocio más crecen estas bases de datos, se fija como *alta*.

- CPLX: Product Complexity

- Baja*, ya que aunque sea un software medianamente grande, no es complejo ni tiene elementos sofisticados.

- RUSE: Required Reusability

- Se pretende que pueda ser adaptable con pocas modificaciones a distintos clientes, por lo que la reusabilidad es *muy alta*.

- DOCU: Documentation Match to Life-Cycle Needs

- Nominal*, no es determinante.

- Factores de la plataforma

- TIME: Execution Time Constraint

- Nominal*, ya que no estamos trabajando con sistemas críticos o similares.

- STOR: Main Storage Constraint

- Nominal* por el mismo motivo.

- PVOL: Platform Volatility

- Nominal*.

- Factores de personal

- ACAP: Analyst Capability

- Ya que los integrantes del grupo tienen conocimientos fuera de la informática también, se considera que tienen una *alta* capacidad de análisis.

- PCAP: Programmer Capability

- Al tratarse de estudiantes todavía no se tiene una gran sapiencia sobre programación, por lo que consideramos esta campo como *bajo*.

- AEXP: Application Experience

- Nominal*.

- PEXP: Platform Experience

- Nominal*.

- LTEXT: Language and Tool Experience

Alta porque trabajamos con Java en un entorno al que estamos acostumbrados: Eclipse. La otra herramienta que utilizamos la mayor parte del tiempo es el TeXworks para escribir en \LaTeX , que también la dominamos bastante.

- PCON: Personnel Continuity

Alta salvo abandono.

■ Factores del proyecto

- TOOL: Use of Software Tools

Bajo ya que apenas se utilizan herramientas, como se ve en el **Plan de proyecto Software**.

- SITE: Multisite development

Bajo, ya que como se ha dicho antes, los integrantes del grupo se ven prácticamente todos los días.

Con estos datos, el COCOMO II determina un factor de ajuste del esfuerzo de 1.19, como se puede ver en la figura.

The screenshot shows a window titled "EAF - Hosteleria" with a close button (X). The window contains a form for inputting COCOMO II rating factors. The form is organized into several sections, each with a label and a grid of input fields for "base" and "Incr%" values.

base + Incr % = rating						
Product:	RELY	DATA	DOCU	CPLX	RUSE	
base	HI	HI	NOM	LO	VHI	
Incr%	0%	0%	0%	0%	0%	
Platform:	TIME	STOR	PVOL			
base	NOM	NOM	NOM			
Incr%	0%	0%	0%			
Personnel:	ACAP	PCAP	PCON	APEX	LTEX	PLEX
base	HI	LO	HI	NOM	HI	NOM
Incr%	0%	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: 1.19

Buttons: OK, Cancel, Help

4. Esfuerzo, coste duración

En la herramienta también se determinó el salario en dólares por mes (que aparece como *LABOR RATE (\$/month)*). Suponiendo un salario humilde, estimamos unos 700

\$/mes. Además, añadimos el lenguaje de programación elegido: Java, como ya hemos dicho en varias ocasiones.

Project Name: Scale Factor: 22.77 Schedule

Project Notes Development Model:

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	Hosteleria	F:14151	700.00	1.19	JAVA	59.9	71.5	197.9	50056.93	3.5	4.9	0.7

Tras todas estas consideraciones, el programa nos da las estimaciones:

Total Lines of Code:	14151	Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Hours/PM:	152.00	Optimistic	57.2	13.7	247.4	40045.54	2.8	4.2	
		Most Likely	71.5	14.7	197.9	50056.93	3.5	4.9	0.7
		Pessimistic	89.4	15.8	159.3	62571.16	4.4	5.6	

SITE: Multisite Development

- **Estimación de esfuerzo:** Según la herramienta, lo más probable es que se necesiten **71.5 personas · mes**, aunque el valor optimista es de 57.2 y el pesimista de 89.4, convirtiéndolo en un proyecto de envergadura media.
- **Estimación de coste:** La estimación del coste, como hemos comentado antes, está hecha con un supuesto de un salario de 700 \$/mes, aunque esto son sólo conjeturas ya que se trata de un proyecto académico. Así, salen **50056.93 dólares**, que son 38.544,637 euros. Además, se puede apreciar que se estima un precio de unos 3.5 \$ por cada instrucción de la aplicación.
- **Estimación de duración:** Según el COCOMO II, durará **14.7 meses**, poco más de un año, entre 13.7 y 15.8 meses en el mejor y peor caso respectivamente. Es interesante ver que si dividimos el esfuerzo entre el tiempo obtenemos el número aproximado de personas que son necesarias cada mes, y en nuestro caso es 4.86, es decir unas 5 personas cada mes, justo el número de miembros del equipo.

Nota: Lo que aparece entre la duración y el coste en la herramienta es la productividad, que se obtiene de dividir SLOC entre el esfuerzo más probable, y obtenemos que es 197.9 SLOC/pm.

