

# Estimación y Planificación

## Software para Cafetería

### *“Café Nostro”*

**Asignatura:**

**Código Asignatura:**

**Profesores:**

**Alumnos:**

Taller de Ingeniería de Software

INFO 264

Raimundo Vega Vega

Cristian Rojas Pérez

Esteban Huenumán Villarroel

Manuel Lavín Brellenthin

Daniel Molina Estefania

Tomas Vera Jackson

# Índice de Contenido

<i>Introducción</i> .....	3
<i>Estimación</i> .....	3
Factor de peso de actores sin ajustar (UAW) .....	3
Factor de peso de los casos de uso sin ajustar (UUCW).....	4
Puntos de casos de uso sin ajustar (UUCP).....	4
Factores técnicos (TF).....	5
Factores ambientales (EF).....	6
Casos de uso ajustado (UCP) .....	6
Esfuerzo (E).....	6
Esfuerzo estimado en horas-persona (E) .....	7
Tiempo.....	7
Esfuerzo meses-hombre .....	7
Esfuerzo por actividad .....	8
<i>Planificación</i> .....	9
Metodología.....	9
Cronograma (Carta Gantt) .....	10
Ciclo de vida.....	11
Roles y asignación de recursos .....	11
Riesgos .....	11

## Introducción

Para realizar la estimación de nuestro proyecto “Café Nostro” se utilizara la métrica Punto de Caso de Uso (UCP). UCP es un método de estimación de esfuerzo para proyectos de software, a partir de sus casos de uso. Con estos datos se conocerá la estimación del tamaño y el esfuerzo invertido en el desarrollo de la funcionalidad especificada en los casos de uso. El método consta de cuatro etapas en las que se desarrollan los siguientes cálculos:

1. Factor de peso de los actores sin ajustar (UAW).
2. Factor de peso de los casos de uso sin ajustar (UUCW).
3. Puntos de caso de uso ajustados (UCP).
4. Esfuerzo horas-hombre.

## Estimación

### Factor de peso de actores sin ajustar (UAW)

Consiste en la evaluación de la complejidad de los actores con los que el sistema podrá interactuar. Este puntaje se calcula determinando para cada actor la forma en que se relacionará con el caso de uso.

A continuación, se muestra la tabla utilizada con los factores que representa a un actor según cada tipo de interacción:

Tipo Actor	Descripción	Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API).	1
Medio	Otro sistema interactuando a través de un protocolo (ej: TCP/IP) o una persona interactuando a través de una interfaz en modo texto.	2
Complejo	Una persona que interactúa con el sistema a través de una interfaz gráfica (GUI).	3

En nuestro caso, los actores que ingresarán al sistema con sus respectivas complejidades, se verá en la siguiente tabla:

Actor	Complejidad	Costo
Dueño	Complejo	3
Administrador	Complejo	3
Cajero	Complejo	3
<b>Total</b>		<b>9</b>

Por lo tanto se obtiene un  $UAW = 9$ .

## Factor de peso de los casos de uso sin ajustar (UUCW)

Este punto funciona muy similar al anterior, pero para determinar el nivel de complejidad se puede realizar mediante dos métodos: basado en transacciones o basado en clases de análisis, en este proyecto nos basaremos en el primero.

Una transacción será definida como un conjunto de actividades atómicas, lo que quiere decir que se ejecutan todas o no se ejecuta ninguna.

A continuación se puede observar una tabla con los tipos de transacciones observadas en el proyecto junto a sus respectivas cantidades y resultados.

Tipo de Caso de Uso	Descripción	Factor
Simple	Menos de 5 clases	5
Medio	De 5 a 10 clases	10
Complejo	Más de 10 clases	15

En la siguiente tabla se detallarán los casos de uso que generaron al momento de planificar el software.

Casos de Uso	Complejidad	Factor
<i>Iniciar sesión</i>	Simple	5
<i>Modificar datos cafetería</i>	Simple	5
<i>Gestión de usuario</i>	Medio	10
<i>Gestionar producto</i>	Medio	10
<i>Registrar venta</i>	Complejo	15
<i>Visualizar estadísticas</i>	Complejo	15
<i>Gestión venta directa</i>	Complejo	15
<i>Gestión pedidos cliente</i>	Complejo	15
<b>Total</b>		<b>90</b>

Por lo tanto, ***UUCW = 90***.

## Puntos de casos de uso sin ajustar (UUCP)

El UUCP son los puntos de casos de uso sin ajustar, el cual sirve para tener una idea más precisa acerca de los casos de uso e interfaces, ya que se toman en cuenta los pesos de los actores y los pesos de los casos de uso.

$$UUCP = UAW + UUCW$$

$$UUCP = 9 + 90$$

$$\mathbf{UUCP = 99}$$

## Factores técnicos (TF)

Este factor se compone de 13 puntos que evalúan la complejidad de los módulos del proyecto que se lleva a cabo, cada uno de estos factores tienen un peso definido con los cuales se obtendrá puntos ponderados por cada uno de ellos, según la valoración que se le asigne. Los factores se deben de evaluar según la tabla siguiente:

Descripción	Valor
<i>Irrelevante</i>	De 0 a 2
<i>Medio</i>	De 3 a 4
<i>Esencial</i>	5

La tabla a continuación da a conocer el valor que se asignó a cada factor y su peso para luego mostrar el resultado.

Factor	Descripción	Valor Asignado	Peso	Resultado
<b>T1</b>	Sistema distribuido	0	2	0
<b>T2</b>	Objetivos de performance o tiempo de respuesta	4	1	4
<b>T3</b>	Eficiencia del usuario final	4	1	4
<b>T4</b>	Procesamiento interno complejo	3	1	3
<b>T5</b>	El código debe ser reutilizable	2	1	2
<b>T6</b>	Facilidad de instalación	5	0,5	2,5
<b>T7</b>	Facilidad de uso	5	0,5	2,5
<b>T8</b>	Portabilidad	3	2	6
<b>T9</b>	Facilidad de cambio	3	1	3
<b>T10</b>	Concurrencia	1	1	1
<b>T11</b>	Incluye objetivos especiales de seguridad	4	1	4
<b>T12</b>	Provee acceso directo a terceras partes	0	1	0
<b>T13</b>	Se requiere facilidades especiales de entrenamiento a usuario	3	1	3
<b>TFactor</b>				<b>35</b>

Por lo tanto, se obtiene que los factores técnicos tienen un equivalente a treinta y cinco.

$$TCF = 0,6 + (0,01 * TFactor)$$

$$TFC = 0,95$$

## Factores ambientales (EF)

Los factores sobre los cuales se realiza la evaluación son 8 puntos, que están relacionados con las habilidades y experiencia del grupo de personas involucradas en el desarrollo del proyecto. Estos factores se muestran a continuación:

Factor	Descripción	Valor	Peso	Resultado
E1	Familiaridad con RUP	0	1,5	0
E2	Experiencia en la aplicación	3	0,5	1.5
E3	Experiencia en orientación a objetos	5	1	5
E4	Capacidad del analista líder	0	0,5	0
E5	Motivación	5	1	5
E6	Estabilidad de los requerimientos	3	2	6
E7	Personal Part-Time	1	-1	-1
E8	Dificultad del lenguaje de programación	2	-1	-2
<b>EFactor</b>				<b>14,5</b>

Por lo tanto, los factores ambientales suman un total de veinte coma cinco puntos.

$$EF = 1,4 + (-0,03 * EFactor)$$

$$EF = 0,965$$

## Casos de uso ajustado (UCP)

El factor resultante de los cálculos anteriores entrega los puntos de caso de uso ajustado, el cual considera los puntos de caso de uso sin ajustar, los factores técnicos y ambientales.

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 99 \times 0,95 \times 0,965$$

$$UCP = 90,758$$

Por lo tanto, se tiene que los puntos de casos de uso ajustados tienen un valor de 73,829.

## Esfuerzo (E)

Este cálculo se realiza con el fin de tener una aproximación del esfuerzo expresado en horas-hombre, pensando solo en el desarrollo según las funcionalidades de los casos de uso.

Primero se debe contar la cantidad de factores ambientales del E1 al E6 que tienen una puntuación menor a 3, también contar la cantidad de estos mismos del E7 y E8 que son mayores que 3

Esfuerzo horas-hombre:

Descripción	Horas /Persona(CF)
Si el valor es $\leq 2$	20
Si el valor es $\leq 4$	28
Si el valor es $\geq 5$	36
<b>CF</b>	20

Según los valores de Factores ambientales de obtiene que valor = 1. Por lo tanto **CF = 20**.

### Esfuerzo estimado en horas-persona (E)

$$E = UCP * CF$$

$$E = 90,758 * 20$$

$$E = 1815,165 \text{ esfuerzo-horas-persona}$$

Por lo tanto, se tiene que el esfuerzo en horas-hombre es de aproximadamente 1815,165.

### Tiempo

Nuestro equipo de trabajo está formado por cuatro integrantes, por lo tanto se tiene que:

$$T = \frac{E}{\text{Integrantes}}$$

$$T = \frac{1815,165}{4} = 453,791 \text{ hrs/hombre}$$

Cada miembro del equipo debe de trabajar 453,791 horas para desarrollar el proyecto.

### Esfuerzo meses-hombre

Asumiendo que un mes tiene 20 días laborales y se trabaja 4 horas al día.

$$E \text{ (m/h)} = 113 \text{ días-hombre} = 5,67 \text{ meses-hombre}$$

Con este análisis logramos observar que aproximadamente el desarrollar este proyecto nos tardaríamos cinco meses aproximadamente si trabajamos los cuatro de una forma pareja.

## Esfuerzo por actividad

UCP tiene su propia estimación de tiempo para las actividades que se desarrollan al momento de desarrollar un software.

Finalmente, el esfuerzo en horas de cada actividad es:

<b>Actividad</b>	<b>Porcentaje</b>	<b>Esfuerzo Horas/Hombre</b>
<i>Análisis</i>	10%	36,9146
<i>Diseño</i>	20%	73,8292
<i>Programación</i>	40%	147,6584
<i>Pruebas</i>	15%	55.3719
<i>Sobrecarga (otras actividades)</i>	15%	55.3719



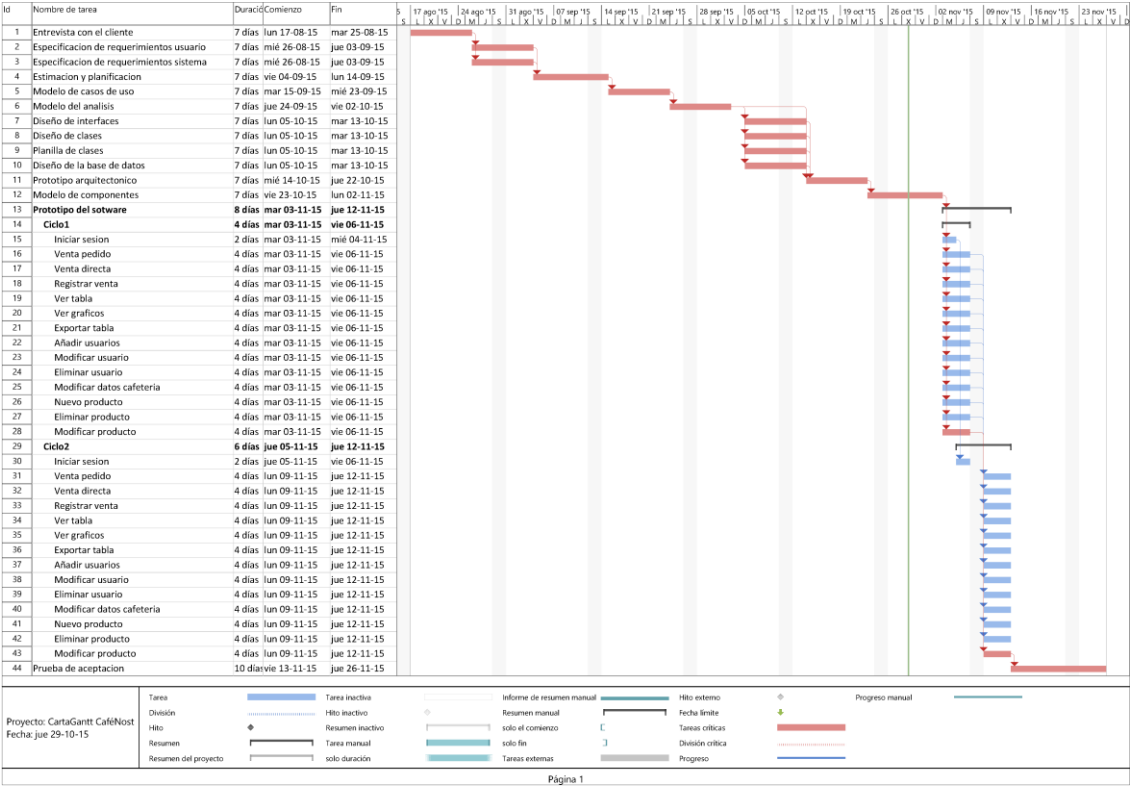
# Planificación

## Metodología

La metodología utilizada es RUP, la cual se divide en 4 fases:

Fase Inicial	Esta fase desarrollará los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en los artefactos URD y Diagrama de procesos. De esta fase se captarán los principales casos de uso para el desarrollo del proyecto. Esta fase da por terminada cuando se capturen todos los requisitos de usuario y se tenga el diagrama de procesos terminado.
Fase Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y/o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera iteración de la construcción deben estar analizados y diseñados (en el modelo de análisis y diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso, la primera iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el modelo de análisis/diseño, también permitirá hacer una revisión general de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos.
Fase de Construcción	Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis/Diseño. El producto se construye en 2 iteraciones, cada una produciendo un entregable al cual se le aplican las pruebas y valida con el usuario. Se comienza la elaboración de material de apoyo al usuario. Esta termina cuando se presenta el entregable operacional al usuario, para poder realizar pruebas beta.
Fase de Transición	En esta fase se prepararán dos entregables para distribución, asegurando una implementación y cambio del sistema previo de manera adecuada, incluyendo la capacitación del usuario, el término de la capacitación y entrega del producto finaliza esta etapa.

Cronograma (Carta Gantt)



## Ciclo de vida

El ciclo de vida utilizado es espiral, el cual organiza las tareas en fases o iteraciones. RUP, como se mencionó en el punto anterior, presenta fases las cuales se descomponen en iteraciones. Si el usuario quiere seguir modificando el software para agregar mejoras, se evalúan las nuevas propuestas, los riesgos que esto trae y se realiza otra vuelta en espiral. Esto puede continuar hasta que el usuario este conforme con el producto de software y no necesite continuar con otro ciclo.

## Roles y asignación de recursos

Los participantes en el proyecto participan en todas las fases, y estará formado por:

**Jefe de Proyecto:** Se encarga de asignar los recursos, gestionar las propiedades, coordinar interacciones con el usuario y mantener al equipo del proyecto enfocado en los objetivos. Además también establece que las prácticas utilizadas aseguren la integridad y calidad de los artefactos del proyecto. El jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema, gestión de riesgos, planificación y control del proyecto.

**Analistas:** Se encargarán de la captura, especificación y validación de los requisitos, interactuando con usuario. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.

**Programadores:** Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario.

**Documentación:** Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas finales, elaboración de la documentación.

## Riesgos

### Diferencias entre el tiempo de desarrollo real del proyecto y la planificación estimada

La planificación orienta todo el desarrollo del proyecto software. Un error en la misma puede incidir directamente en sus resultados. Este problema puede surgir debido a que el grupo de trabajo posee poca experiencia en tareas de planificación.

### Requisitos del producto software poco claros

Al realizar la consulta al usuario, este no sabe expresar concretamente qué servicios espera obtener con su aplicación. Los requisitos representan la idea que tiene el usuario sobre su producto, sobre ellos se construyen los casos de uso y dichos casos de uso orientan el desarrollo del proyecto. Una mala recolección de los mismos afectaría la calidad de todo el proyecto software. La incorporación o

modificación de requisitos durante el desarrollo del proyecto requeriría realizar cambios sobre la documentación de la aplicación elaborada con anterioridad.

### **Error en el diseño del proyecto software**

Al realizar las tareas de implementación puede encontrarse que el diseño carece del suficiente nivel de detalle o está mal enfocado, teniendo como consecuencia, que la aplicación no cumpla con las expectativas que el cliente espera y con ello, tener que modificar los artefactos de diseño.

### **Falta de Experiencia con las herramientas a utilizar en el proyecto software**

El equipo de desarrollo tiene dificultades a la hora de realizar sus tareas, ya sea en documentación como en implementación, debido a su inexperiencia con las herramientas disponibles para el proyecto software. Debido a esto, se deberá invertir tiempo extra en aprender el funcionamiento de dichas tecnologías de desarrollo.

### **Abandono temporal de un miembro del equipo de desarrollo**

Algún miembro del equipo de desarrollo no se encuentra disponible por cualquier motivo externo (por ejemplo: enfermedad.) durante un periodo corto de tiempo, y por lo tanto no puede realizar tareas relacionadas con el proyecto software. La falta de disponibilidad de los recursos humanos puede provocar el retraso con respecto a la planificación inicial de cualquier actividad del proyecto.

### **Falta de comunicación entre los integrantes del equipo de desarrollo**

Durante la realización de un proyecto software, hay muchos artefactos que realizar y tareas que completar por la totalidad de los integrantes del equipo de desarrollo. Normalmente dichas tareas están relacionadas de alguna manera, y cualquier cambio independiente en una de ellas afecta al resultado final o a otras tareas.

### **Conflictos entre los integrantes del equipo de desarrollo**

Los conflictos pueden surgir debido a la aparición de problemas o discrepancias entre los miembros del equipo de desarrollo. Si los desacuerdos no son zanjados rápidamente, se pueden provocar retrasos teniendo como consecuencia que, tener que reajustar la planificación del proyecto software.