

# Técnica híbrida de estimación basada en el análisis de puntos de función y puntos de casos de uso

## Hybrid estimation technique based on the analysis of function points and use cases points

Cesar Yesid Barahona Rodríguez<sup>1</sup>, Daniel Sebastián Arias Rojas<sup>2</sup>, Paola Andrea Chia Rodríguez<sup>3</sup>  
<sup>1,2,3</sup> Universidad de Cundinamarca

<sup>1</sup> cbarahona@ucundinamarca.edu.co, <sup>2</sup> danielarias@ucundinamarca.edu.co, <sup>3</sup> pchia@ucundinamarca.edu.co

**Resumen**— La estimación de software es una etapa fundamental a la hora de desarrollar un aplicativo web o de escritorio, ya que con esta fase se podrá tener una idea del tiempo, costo y personal que se debe utilizar para finalizar sin contratiempos un proyecto. Es por eso que la Universidad de Cundinamarca, sede Facativá ha querido desarrollar un proyecto de investigación e ingeniería que se base en el estudio de dos diferentes técnicas de estimación como lo son Puntos de función y Puntos de Casos de Uso y gracias a este análisis desarrollar una técnica híbrida que abarque las ventajas de cada una de los métodos planteados.

**Palabras claves**— Estimación, esfuerzo, desarrollo, costo, casos de uso, requerimientos.

**Abstract**— Software estimation is a fundamental step when developing a web or desktop application, because with this phase you can get an idea of the time, cost and personnel that should be used to finish a project without mishaps. That is why the University of Cundinamarca, sectional Facultative has wanted to develop a research and engineering project that is based on the study of two different estimation techniques such as Function Points and Use Case Points and thanks to this analysis develop A hybrid technique embracing the advantages of each of the methods proposed.

**Keywords**— Estimation, effort, development, cost, use cases, requirements.

### 1. Introducción

Al hablar sobre la estimación de esfuerzo, duración y costos de proyectos de software es muy común encontrar diversos métodos que tratan de una u otra manera hallar solución a esta necesidad, la cual hoy por día causa riesgo en algunas por no decir todas empresas de software, esto es debido a que fracasan en su desarrollo por el aumento de costos en la producción del mismo, falta de personal o de tiempo; causando así que el producto final sea deficiente o también se termina abandonando el proyecto [1], sin embargo, hay que aclarar que no es la única causa de fracaso en los proyectos de software.

Existen muchos métodos de estimar proyectos de software unos más eficaces que otros y adecuados para cada situación; por esa razón este proyecto se enfoca en dos metodologías: Puntos de Casos de uso y Puntos de Función, las cuales se tomaron como punto de partida para el estudio o formación de un híbrido entre las dos, el

cual busca unir y complementar las fortalezas de cada técnica:

- **Puntos de casos de uso:** Desarrollado por Gustav Karner en 1993. Este método utiliza los actores y casos de uso que se plantean en una de las etapas de ingeniería de software del proyecto a desarrollar. A los casos de uso se les asigna una complejidad basada en transacciones (interacción usuario-maquina), mientras que a los actores se les asigna una complejidad basada en su tipo. Este método se escogió en esta investigación ya que quiso dar importancia a los requerimientos determinados por el cliente o usuario (los estudiantes en el caso de la universidad de Cundinamarca) y así mismo dándole una gran importancia a la etapa de modelamiento de software y conectándola a la fase de estimación.

- **Puntos de Función:** Desarrollado por Allan Albrecht a mediados de la década de los 70's y posteriormente modificado y mejorado de forma continua por la International Function Point Users Group (IFPUG), este método está basado en la norma ISO/IEC 20926:2009 la cual se tomó como base para entender a fondo como se debe realizar dicha estimación con sus respectivos estándares oficiales.

Al tomar estas dos metodologías, se procedió a identificar las fortalezas de cada una, puesto que con ellas se quiso complementar y hallar un punto en común entre estas, para posteriormente crear un híbrido aproximado (ya que no se puede asegurar que sea la mejor opción de estimación o si por el contrario es mejor utilizar una técnica por separada) entre estas, la razón para esto es simple, se quiso plantear una nueva opción que combata o reduzca la incertidumbre en estas técnicas y no solo eso sino en el resto de métodos de estimación de proyectos que se encuentran actualmente, tratando así de lograr mejores resultados.

## 2. Planteamiento del problema

Se conoce que a la hora de saber la duración y el costo de un proyecto de software se tiene que llevar a cabo una debida estimación para que durante el proceso de diseño, construcción y entrega no ocurran pérdidas de tiempo y dinero que vaya a perjudicar a la empresa a futuro y no solo a la empresa a la relación y cumplimiento con el cliente.

Asimismo, se tiene planteado la posibilidad de crear un método híbrido basándose en las fortalezas de dos métodos de estimación muy importantes y reconocidos en el ámbito empresarial y académico como lo son Puntos de función y Puntos de casos de uso, en donde se quiere realizar un comparativo entre los resultados arrojados por las tres técnicas (incluyendo las dos anteriores y el híbrido) y poder ver cuál es la mejor opción a la hora de limitar y escatimar tiempo y costos teniendo siempre como base que esto será aproximado ya que no se puede saber a precisión cuál de los tres será superior a la hora de reducirlos.

Es importante sustentar el porqué de la idea de trabajar con estos dos métodos de estimación ya mencionados y un híbrido creado con las características de los dos anteriores, una de los motivos se debe a lo planteado por el profesional José PowSang, docente de la Universidad Católica del Perú en el artículo "Estudio de Técnicas

basadas en Puntos de Función para la Estimación del Esfuerzo en Proyectos Software" en el cual menciona que "el análisis de puntos de función se puede aplicar de manera sencilla con los casos de uso mejorando la calidad de los documentos de requerimientos y, a la vez, mejorando la estimación del proyecto de software. El aplicar la técnica de puntos de función permite verificar y validar el contenido de un documento de especificación de Requisitos de software" [2].

## 3. Análisis comparativo métodos de estimación

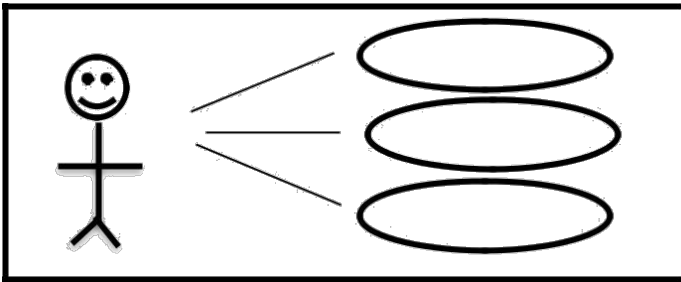
En este análisis comparativo sobre la estimación de proyectos de software se profundizará sobre dos métodos que son: Puntos de Casos de uso y Puntos de Función.

Una ventaja de hacer el análisis comparativo entre estas dos técnicas de estimación es que contienen la misma estructura, lo cual facilita la realización del híbrido.

### 3.1. Estimación por Puntos de Casos de uso UCP

El método de Puntos de casos de uso para la estimación de proyectos de software se adecua a la idea que se quiere plasmar en la investigación de darle una importancia vital a los requerimientos y al modelado de software en su diagrama de casos de uso, para hablar de las características principales de este método se toma un fragmento del artículo de investigación de los ingenieros de la universidad Nacional del nordeste de Argentina, el cual menciona lo siguiente: "UCP (Use cases Point) se basa en la utilización de casos de uso como dato de entrada para calcular el esfuerzo requerido, en horas-hombre (hh), para el desarrollo de un proyecto de software. El método utiliza cuatro variables principales: Actores, Casos de uso, Factores técnicos de complejidad, Influencias técnicas que puedan afectar el proceso de desarrollo" [3]. Estos actores que son mencionados en el artículo de investigación son los plasmados en el diagrama de casos de uso del modelado UML es por eso que quien esté a cargo de la fase de ingeniería de software no debe pasar por alto ninguna funcionalidad en los diagramas a realizarse.

Para la estimación a través de este método Puntos de Casos de Uso, se hace un análisis del modelo UML.



**Figura 1.** Diagrama de casos de uso.

Como se ilustra en la *Figura 1. Diagrama de casos de uso*, el modelo de casos de uso se basa en dos importantes componentes como lo son: el actor que representa al usuario o sistema que tiene contacto con el software, y los casos de uso son las funcionalidades que dispone el actor.

Este método se compone de dos fases principales en las cuales evaluará funcionalidades y componentes del software a desarrollarse.

### 3.1.1. Evaluación de funcionalidades del software a través de UCP

Se denominan también como Puntos de casos de uso sin ajustar. Para desarrollar los UUCP, se dividen en dos ítems denominados:

- Factor de peso de los actores sin ajustar (UAW)
- Factor de peso de los casos de uso sin ajustar (UUCW)

En donde,

- Los UAW o factor de peso de los actores sin ajustar se basa en la cantidad de actores presentes en el sistema y en la complejidad que se le sea asignada. La complejidad de los actores se determina si se trata de una persona o de otro sistema quien interactúa con el software.
  - Actor simple: Otro sistema que interactúa con el software a desarrollar mediante una interfaz de programación (API).
  - Actor Promedio: Otro sistema que interactúa con el software a desarrollar mediante un protocolo (ejemplo: mediante TCP/IP).

- Actor Complejo: Una persona que interactúa con el sistema mediante una interfaz gráfica.

- Los UUCW o factor de peso de los casos de uso sin ajustar se realiza un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. Esta complejidad se lleva a cabo mediante la cantidad de transacciones en el caso de uso.

- Casos de uso simple: Es aquel caso de uso que tiene de 1 a 3 transacciones.
- Casos de uso promedio: Es aquel caso de uso que tiene de 4 a 7 transacciones.
- Casos de uso complejo: Es aquel caso de uso que tiene más de 8 transacciones.

### 3.1.2. Factores técnicos de ajuste

El cálculo de factores de ajuste que trabaja el método de Puntos de Casos de uso (UCP) consta de dos etapas, la primera es el cálculo del Factor de complejidad técnica (TCF) en donde se realiza una cuantificación de un conjunto de factores que pueden o no influir en el sistema a desarrollar, cada uno de estos se califica de 0 a 5 donde, 0 es un aporte irrelevante y 5 un aporte muy importante.

#### Factores:

- Sistema distribuido
- Tiempo de respuesta
- Eficiencia por el usuario
- Proceso interno complejo
- Reusabilidad
- Facilidad de instalación
- Facilidad de uso
- Portabilidad
- Facilidad de cambio
- Concurrencia
- Objetivos especiales de seguridad
- Acceso directo a terceras partes
- Facilidades especiales de entrenamiento a usuarios finales

La segunda etapa del factor técnico en este método de estimación, es el cálculo del factor ambiental (FE), el cual mide las habilidades y el entrenamiento del grupo involucrado en el desarrollo, por eso es tan importante esta parte en la estimación por puntos de casos de uso, su

calificación es similar al de TCF; se le debe asignar una cuantificación a un conjunto de factores.

#### **Factores:**

- Familiaridad con el modelo del proyecto usado
- Experiencia en la aplicación
- Experiencia en POO
- Capacidad del analista líder
- Motivación
- Estabilidad de los requerimientos
- Personal de media jornada
- Dificultad en lenguaje de programación

### **3.2. Estimación por Puntos de función PF**

Este método de estimación tiene una gran sustentación para poder ser aplicado en cualquier proyecto gracias al respaldo de la norma ISO/IEC 20926:2009 [4].

Una razón muy válida del porque usar el método de puntos de función para la estimación de software lo da la profesional Gabriela Salazar en su artículo “Estimación de proyectos de software: un caso práctico” en el cual menciona lo siguiente: “Los Puntos de Función (PF) son una medida del tamaño de una aplicación desde el punto de vista funcional o del usuario. Este es un método que sistemáticamente se puede aplicar a cualquier tipo de software, ya sea en desarrollo o mantenimiento. Está basado en la funcionalidad del software y su complejidad. Los PF son derivados de aspectos externos de las aplicaciones de software como: entradas, salidas, consultas, archivos lógicos e interfaces. El uso de los PF ha ganado aceptación como una medida de productividad clave y los procedimientos de conteo han sido actualizados varias veces desde su primera publicación. El FPA es ahora administrado por el International Function Point Users Group (IFPUG). Ellos proveen los estándares para aplicar el cálculo de los PF a través de su publicación Counting practices manuals” [5].

Para llevar a cabo la estimación de software a través del método de puntos de función se deben llevar a cabo dos etapas fundamentales, así como en el método de Puntos de casos de uso denominadas Puntos de función sin ajustar (Evaluación de funcionalidades) y Puntos ajustados (Evaluación de factores técnicos).

#### **3.2.1. Evaluación de funcionalidades**

En este método se denominan como Puntos de función sin ajustar (PFSA).

Para desarrollar los PFSA, se clasifican los requerimientos en cinco tipos para tener un mejor control sobre ellos:

- EI (Entrada Externa)
- EO (Salida Externa)
- EQ (Consulta Externa)
- ILF (Archivo Lógico Interno)
- EIF (Archivo de Interfaz Externo)

En donde,

- “EI” equivalen al número de interfaces o pantallas en donde el usuario ingresa datos al sistema. Se puede clasificar en complejidad baja (3) media (4) y alta (6).

- “EO” se refiere al número de informes, gráficos y listado de datos del sistema, Se puede clasificar en complejidad baja (4) media (5) y alta (7).

- “EQ” son las búsquedas de datos que realizan los usuarios del software (ejemplo: buscadores), Se puede clasificar en complejidad baja (3) media (4) y alta (6).

- “ILF” Es un grupo de datos relacionados, tal como los percibe el usuario y que son mantenidos por la aplicación. Cabe aclarar que los ficheros se cuentan una sola vez, independientemente del número de procesos que los acceden. (Ejemplo base de datos, servidor FTP), Se puede clasificar en complejidad baja (7) media (10) y alta (15).

- “EIF” se refiere a un grupo de datos relacionados, tal como los percibe el usuario, referenciados por la aplicación y que son mantenidos por otra aplicación. Cabe resaltar que son ficheros internos de otra aplicación (ejemplo WEB SERVICES), Se puede clasificar en complejidad baja (5) media (10) y alta (10).

#### **3.2.2. Factores técnicos de ajuste**

El cálculo del Factor de ajuste que se basa en la calificación de un grupo de factores de ajustes el cual será evaluado de 0 a 5 donde 0 es un nivel de influencia irrelevante y 5 uno muy importante.

### Factores:

- Comunicación de datos
- Procesamiento distribuido
- Objetivos de rendimiento
- Configuración del equipamiento
- Tasa de transacciones
- Entrada de datos en línea
- Interfaz con el usuario
- Actualizaciones en línea
- Procesamiento complejo
- Reusabilidad del código
- Facilidad de implementación
- Facilidad de operación
- Instalaciones múltiples
- Facilidades de cambio

### 3.3. Similitudes y ventajas de estos métodos

Al conocer la estructura de los dos métodos de estimación propuestos para la realización del método híbrido, una etapa fundamental para este análisis es poder conocer las similitudes y las ventajas de cada una de estas técnicas.

#### 3.3.1. Similitudes

- Los métodos de estimación Puntos de Casos de Uso (UCP) y Puntos de Función (PF) contienen la misma estructura de análisis de datos referentes al proyecto. Ambas técnicas trabajan dos fases conocidas como “Valores sin Ajustar” y “valores ajustados”.

- Para pasar de los “Valores sin Ajustar” a los “Valores ajustados” ambos califican “factores técnicos” que consisten en factores externos o internos que pueden afectar o favorecer al desarrollo del software.

- Ambos métodos califican funcionalidades del software, pero específicamente PF evalúa requerimientos funcionales y UCP los casos de uso de los diagramas UML.

#### 3.3.2. Ventajas

- UCP además de evaluar funcionalidades y su complejidad, evalúa los actores.

- Los factores de ajuste de UCP tienen en cuenta los factores de ambiente que influyen en el software.

- PF tiene una norma ISO/IEC en donde explica de forma detallada la forma de evaluar los requerimientos.

- PF Es la base de otros métodos de estimación como Puntos de casos de uso.

- UCP “trabaja bien con diferentes tipos de software”. [6]

- UCP “Muestra buen rendimiento en proyectos pequeños, medianos y grandes”. [6]

### 4. Método híbrido

El método híbrido de estimación de software surgió tras la idea de asociar dos importantes técnicas conocidas a nivel mundial como lo son el análisis de Puntos de Función (PF) y Puntos de Casos de Uso (UCP).

Ambas técnicas de estimación se basan en las calificaciones de funcionalidades, PF directamente evalúa la lista de requerimientos funcionales recogidas por el equipo de desarrollo y UCP en cambio califica el diagrama de Casos de uso. Estas evaluaciones se realizan mediante la experiencia de los usuarios dada en la complejidad (alta, media o baja) que le den a cada una de las funcionalidades.

Los puntos de Casos de Uso (UCP) y los Puntos de Función tienen la misma estructura, lo cual es una gran ventaja a la hora de crear el método híbrido. Esta estructura se divide en tres fases:

- Valores Sin Ajustar
- Valores Ajustado
- Estimación del Esfuerzo

La construcción del método híbrido de estimación se desarrolló en varias etapas denominadas:

- Construcción de la fase Híbrido Sin ajustar
- Construcción del factor de ajuste del método Híbrido
- Cálculo del valor en horas por cada Punto ajustado del híbrido

#### 4.1. Construcción de la fase Híbrido Sin ajustar

La fase “Híbrido Sin ajustar” es la mezcla de “Puntos de Función sin ajustar” y “Casos de Uso Sin ajustar” ya que

se tiene en cuenta las funcionalidades evaluadas por el método UCP como lo son los actores y los casos de uso, pero los parámetros de calificaciones y ponderación de la complejidad son de PF.

Para ser más precisos, la evaluación de los actores no varía con respecto a UCP.

Tabla 1. Calificación Actores.

Actor	Factor de Peso (P)	Número de Actores (N)	Resultado
Simple	1	Ns	P * Ns
Medio	2	Np	P * Np
Complejo	3	Nc	P * Nc
Calificación Actores=			$\sum P*N$

La evaluación de las funcionalidades será la siguiente:

- Determinar tipo de cada Caso de Uso:

Tabla 2. Clasificación de los casos de uso.

Tipo de Caso de Uso	Baja	Media	Alta
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
EIF	6	7	10
Calificación Funcionalidades =			$\sum (\# \text{Casos de Uso} * \text{Complejidad})$
Hibrido Sin Ajustar = Calificación Actores + Calificación Funcionalidades			

- La complejidad del caso de Uso estará dada por la cantidad de transacciones con las que cuenta:

Tabla 3. Determinar Complejidad Caso de Uso.

Complejidad	Descripción
Baja	De 1 a 3 transacciones
Media	De 4 a 7 transacciones
Alta	8 o más transacciones

#### 4.2. Construcción de la fase Factor de ajuste

El factor de ajuste es un valor determinado por la calificación de elementos influyentes en el software, en

PF se conocen como “Factor de Ajuste” y en UCP se dividen en dos tipos: “Factores de Complejidad Técnica” y “Factores de Ambiente”.

La idea fundamental del método híbrido desarrollado es tener en cuenta todos estos factores y la evaluación que el usuario ha dado en la estimación por PF y UCP, es por eso que en esta técnica híbrida se ha desarrollado un término denominado “Porcentaje de Variación ponderado (% ponderado)” que se calcula entre los resultados sin ajustar y ajustados de cada uno de los resultados de las estimaciones realizadas por cada método (ver Tabla 4. Cálculo del % de variación ponderado (híbrido)).

Tabla 4. Cálculo del % de variación ponderado (híbrido).

Método	Valor Sin Ajustar	Valor Ajustado
Puntos de Función	PFSA	PFA
Puntos de Casos de Uso	UUCP	UCP
Resultado Ponderado (RP)	RPSA = PFSA + UUCP	RPA = PFA + UCP
Porcentaje de Variación Ponderado	$\% \text{ ponderado} = ((RPA - RPSA) / RPSA)$ <b>Puntos ajustados del híbrido =</b> Puntos sin ajustar del híbrido + (Puntos sin ajustar del híbrido * %ponderado)	

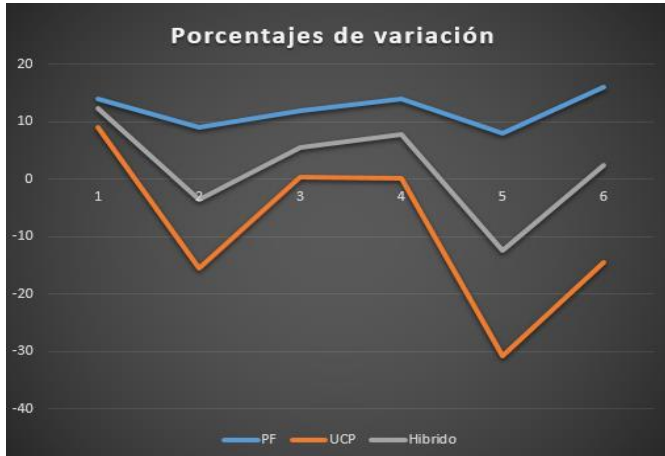
Se realizaron pruebas con diferentes proyectos de software en la Universidad de Cundinamarca para observar los resultados del híbrido. Como se observa en la figura 1, se presenta una comparación gráfica del porcentaje de variación entre puntos sin ajustar y puntos ajustados de los tres métodos de estimación planteados.

#### 4.3. Cálculo del valor en horas por cada punto ajustado del híbrido y estimación de esfuerzo

El valor estándar en horas de cada Puntos de Función ajustado está establecido en 8 horas por cada uno de ellos, y en Puntos de Casos de Uso su valor esta dado en 20 horas.

Para calcular la cantidad de horas de cada punto ajustado del híbrido se creó un promedio ponderado en donde se tenga en cuenta los resultados de las estimaciones del

proyecto evaluado (ver Tabla 5. Calculo del promedio ponderado (Cantidad de horas \* hibrido ajustado)).



**Figura 1.** Comparativa de porcentajes de variación entre métodos.

En los proyectos de prueba estimados de la Universidad de Cundinamarca se presentaron los resultados de la **Tabla 6. Resultados de los proyectos**, por los diferentes métodos.

**Tabla 5.** Calculo del promedio ponderado (Cantidad de horas \* hibrido ajustado).

Método	Valor Ajustado	Estimación
Puntos de Función	PFA	$E_{PF} = PFA * 8$
Puntos de Casos de Uso	UCP	$E_{UCP} = UCP * 20$
Resultado Ponderado (RP)	$RPA = PFA + UCP$	$E_{RP} = E_{PF} + E_{UCP}$
Promedio Ponderado (Cantidad de Horas por Puntos Ajustados del hibrido)	<b>Promedio ponderado</b> = $E_{RP} / RPA$ <b>Estimación de esfuerzo por Híbrido</b> = Puntos Ajustados del hibrido * Promedio Ponderado.	

**Tabla 6.** Resultados de los proyectos.

#	E_PF horas	En E_UCP horas	En Estimación de hibrido (en horas)
1	893,76	1222	425,89974
2	1979,44	3950	2160,8371
3	698,88	2048	1083,4915
4	702,24	1282	464,39659
5	587,52	1052	558,64353
6	853,76	1266	573,65353

## 5. Conclusiones

- Actualmente los métodos de estimación son muy importantes para toda empresa a la hora de disminuir costos, esfuerzos y duración de un proyecto a realizarse.
- Llevar a cabo una correcta estimación no es fácil ya que no todos los métodos arrojaran el mismo resultado; por ende, se debe tener en cuenta la opinión de personas expertas en el campo. Igualmente, en estos métodos siempre habrá un grado de incertidumbre el cual se refiere a que no es exacto en total magnitud.
- Gracias al estudio realizado se ha identificado que los métodos de estimación de Función y puntos de casos de uso se pueden complementar para realizar una técnica hibrida ya que cuentan con una estructura similar al momento de aplicar sus fórmulas.
- Los puntos de función se pueden aplicar de manera sencilla con los casos de uso mejorando la calidad de los documentos de requerimientos, mejorando la estimación del proyecto de software. El aplicar la técnica de puntos de función permite verificar y validar el contenido de un documento de especificación de requerimientos.

## 6. Reconocimiento

Se le agradece a la Universidad de Cundinamarca por generar espacios para fomentar la innovación tecnológica por parte de los estudiantes de Ingeniería de sistemas y carreras afines.

Se les agradece a los diferentes entes educativos por crear eventos para exponer los proyectos creados por los estudiantes de diferentes universidades de Colombia y motivar a la creación de diferentes ideas innovadoras.

## **7. Referencias**

- [1] G. N. Dapozo, Y. Medina, A. B. Lencina, and G. O. Pedrozo, “Métodos de estimación de esfuerzo y duración en proyectos web pequeños,” 2014.
- [2] A. P. Portillo, “Estudio de Tecnicas basadas en Puntos de Funcion para la Estimacion del Esfuerzo en Proyectos Software,” 2004.
- [3] G. N. Dapozo, M. de los Á. Ferraro, Y. Medina, G. Petrazzini Pedrozo, B. Lencina, and E. Irrazábal, “Estimación de software: métodos orientados a la gestión ágil de proyectos, desarrollo web y estimación temprana,” p. 5 pag., 2015.
- [4] IFPUG, Norma ISO/IEC 20926 - Software and systems engineering -- Software measurement -- IFPUG functional size measurement method, 2009.
- [5] G. Salazar-B, “Estimación de proyectos de software: un caso práctico,” *Ing. y Cienc. - ing.cienc.*, vol. 5, no. 9, pp. 123–143, 2009.
- [6] S. Valero Orea, “Estimación de proyectos de software con puntos de casos de uso,” p. 9, 2010.