**UNIVERSIDAD DEL QUINDÍO  
FACULTAD DE INGENIERÍA**

**INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**INGENIERIA DE SOTWARE III**

**Paulo Cesar Alvis Ramos, Diego Alejandro Sánchez Parra, Leidy Andrea Raigoza Palacio**

**INTRODUCCIÓN**

**En el presente documento encontrará la especificación d**el método de estimación QUELCE (SEI) aplicado al proyecto “Implementación de un prototipo basado en realidad virtual para la enseñanza de cálculo multivariado y vectorial para estudiantes de Ingeniería de Sistemas y Computación de la Universidad del Quindío”, con los pasos definidos en el flujo de análisis, así como los resultados obtenidos. Adicional encontrará la estimación del proyecto aplicando la estrategia Test Driven Development (TDD).

1. **FACTORES DE CAMBIO**

Tabla 1: Matriz de factores de cambio que permite visualizar la probabilidad y el grado de impacto que genera del factor correspondiente.

|  |  |  |
| --- | --- | --- |
| Factor | Probabilidad que ocurra | ¿Qué tanto cambia el programa? |
| Limitaciones respecto a los plugins que pueden ser utilizados en la plataforma Unity 3D. | 25% | Considerablemente |
| Una mala experiencia de usuario al interactuar con el prototipo. | 25% | Medianamente |
| Incompatibilidad entre los modelos 3D construidos en la plataforma Blender y su montaje en Unity 3D. | 25% | Completamente |
| La no comunicación oportuna entre los Stakeholders que permita definir un alcance claro sobre las funcionalidades del prototipo. | 15% | Medianamente |

1. **CHANGE DRIVERS Y MATRIZ DE ESTRUCTURA DE DISEÑO (DSM)**

**Para el proyecto se identifican los Change Drivers (factores de cambio) de la siguiente manera:**

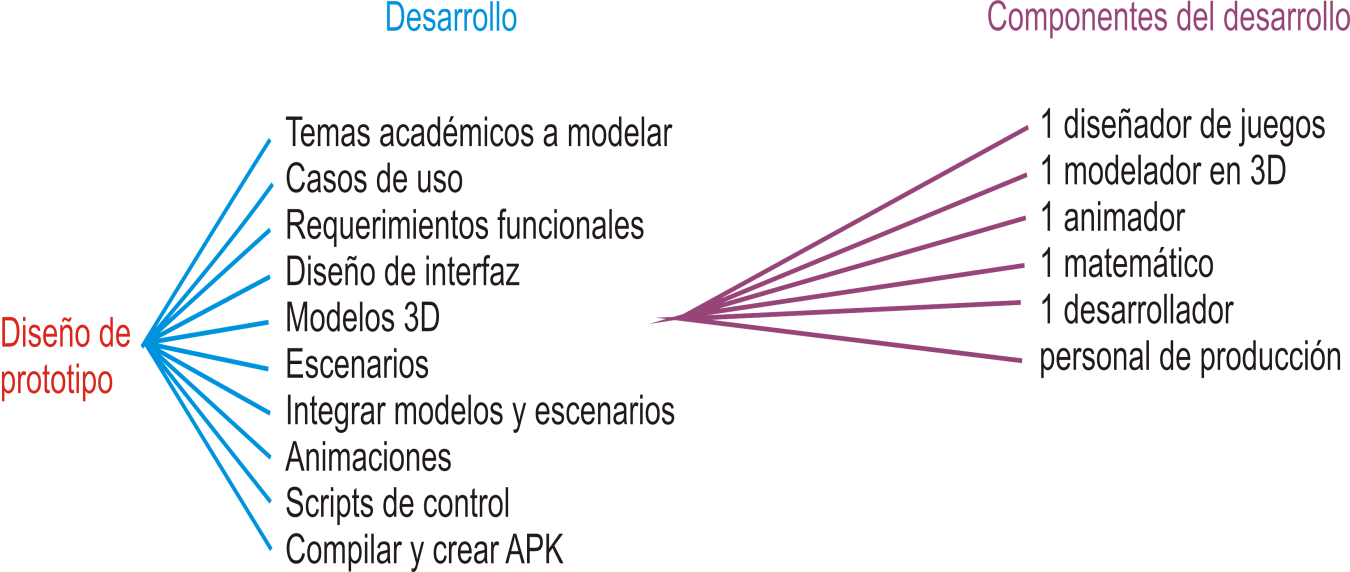


Imagen 1: Imagen que permite visualizar el diseño del prototipo con los puntos de desarrollo

El siguiente paso fue tener en cuenta en la distribución de pesos por factor de cambio, que en nuestro proyecto es la siguiente:

3 puntos = poca influencia

6 puntos = influencia moderada

9 puntos = alta influencia

Ahora creamos la matriz de dependencias (DSM):

Tabla 2: Tabla que permite identificar la matriz de dependencia

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Requisito** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** |
| **A** | **Temas académicos a modelar** | **A** |  |  |  |  |  |  |  |  |  |
| **B** | **Casos de uso** |  | **B** |  |  |  |  |  |  |  |  |
| **C** | **Requerimientos funcionales** |  |  | **C** |  |  |  |  |  |  |  |
| **D** | **Diseño interfaz** |  | **3** | **9** | **D** |  |  |  |  |  |  |
| **E** | **Modelos 3D** | **9** |  |  | **9** | **E** |  |  |  |  |  |
| **F** | **Escenarios** | **3** |  |  | **9** |  | **F** |  |  |  |  |
| **G** | **Integrar modelos y escenarios** |  | **6** |  | **6** | **9** | **9** | **G** |  |  |  |
| **H** | **Animaciones** | **6** |  |  | **3** | **3** |  |  | **H** |  |  |
| **I** | **Scripts de control** |  | **9** | **6** |  | **3** |  |  | **6** | **I** |  |
| **J** | **Compilar y crear APK** |  |  | **3** |  |  | **6** | **9** |  |  | **J** |

1. **PROBABILIDADES CONDICIONALES**

Las diferentes probabilidades Condicionales para cada nodo y el total, son las siguientes:

Tabla 3: Matriz que permite visualizar la probabilidad condicional de cada nodo

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Probabilidad Condicional | Probabilidad de ocurrencia | | | | | | | Total | |
| P (A) | 0,33 | = |  |  |  |  |  | | 33% |
| P (B) | 0,33 | = |  |  |  |  |  | | 33% |
| P (C) | 0,33 | = |  |  |  |  |  | | 33% |
| P (D) | 0,2 | 0,3 | 0,9 | = | **1,4** |  |  | | 14% |
| P (E) | 0,2 | 0,9 | 0,9 | = | **2** |  |  | | 20% |
| P (F) | 0,2 | 0,3 | 0,9 | = | **1,4** |  |  | | 14% |
| P (G) | 0,4 | 0,6 | 0,6 | 0,9 | 0,9 | = | **3,4** | | 34% |
| P (H) | 0,3 | 0,6 | 0,3 | 0,3 | = | **1,5** |  | | 15% |
| P (I) | 0,4 | 0,9 | 0,6 | 0,3 | 0,6 | = | **2,8** | | 28% |
| P (J) | 0,3 | 0,3 | 0,6 | 0,9 | = | **2,1** |  | | 21% |

1. **ESTIMACIÓN DE CADA DRIVER**

Tabla 4: Estimación de cada actividad o driver

|  |  |  |
| --- | --- | --- |
| **Id** | **Requisito** | **Estimación** |
| **D** | Diseño interfaz | 14% |
| **E** | Modelos 3D | 20% |
| **F** | Escenarios | 14% |
| **G** | Integrar modelos y escenarios | 34% |
| **H** | Animaciones | 15% |
| **I** | Scripts de control | 28% |
| **J** | Compilar y crear APK | 21% |

1. **RED BAYESIANA**

Ahora podemos construir el grafo de acuerdo a la matriz generada (BBN):

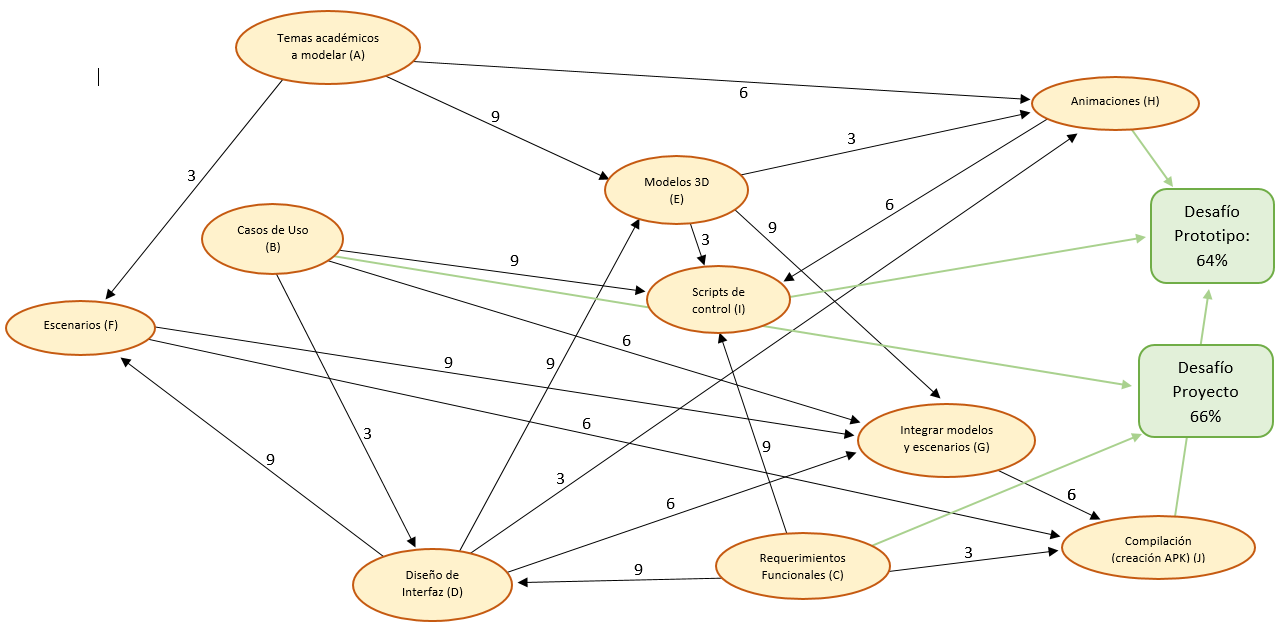


Imagen 2: Grafo que representa la red Bayesiana

Como vemos, básicamente los drivers apuntan a afectar dos áreas: Proyecto y Prototipo. Los nodos de resultado (en verde) reflejan las probabilidades en el contexto de todos los estados posibles de todos los nodos driver.

1. **ESTIMACIÓN DE COSTO**

Ahora aplicamos las salidas que desafían el Prototipo y el Proyecto como entradas en la herramienta de estimación COCOMO II.

* 1. **EN PROYECTO:**

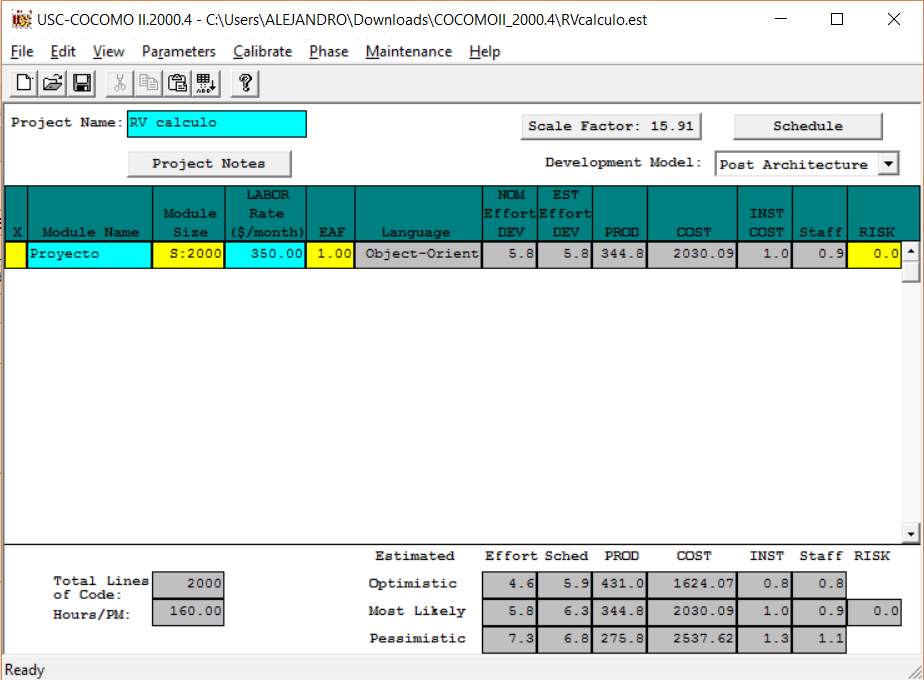


Imagen 3: Captura de la herramienta COCOMO con el cual se realizó la estimación en Proyecto.

* 1. **EN PROTOTIPO:**

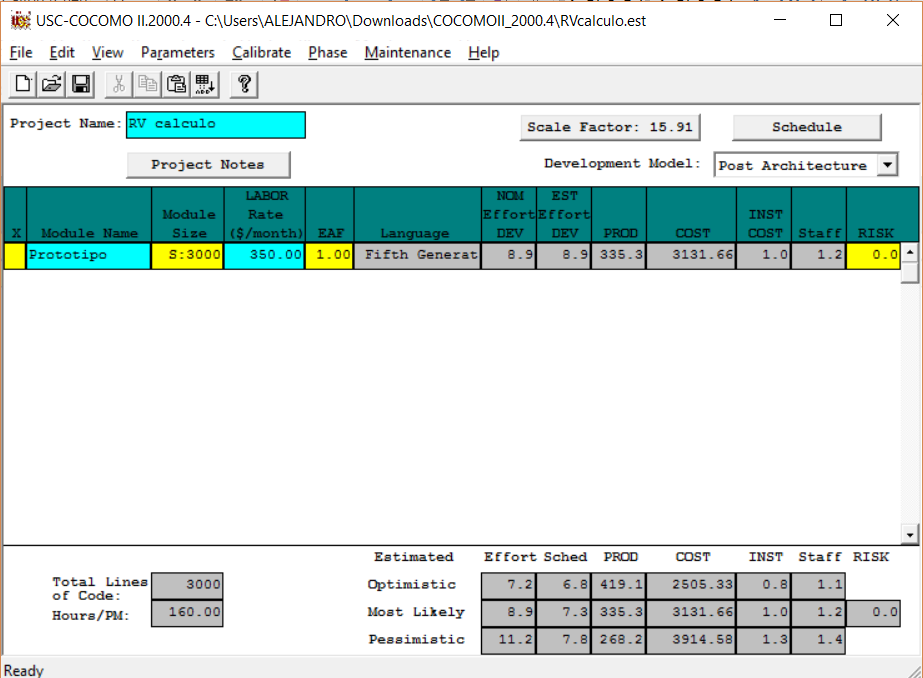


Imagen 4: Captura de la herramienta COCOMO con el cual se realizó la estimación en Prototipo.

* 1. **COSTO MONETARIO CON MONTE CARLO:**

Con los valores en los tres casos: pesimista, ideal y optimo, podemos aplicar las simulaciones con la herramienta CRYSTAL BALL DE ORACLE. Así obtenemos la estimación por costo o por esfuerzo mensual:

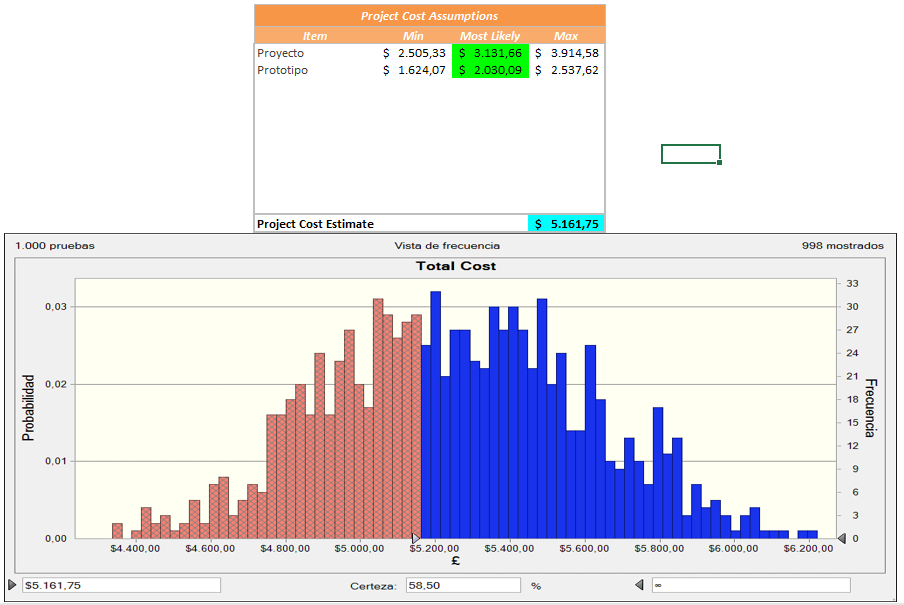


Imagen 5: Imagen que permite visualizar la estimación por costo o esfuerzo mensual.

De esta manera podemos concluir que nuestro proyecto tendrá un costo total estimado en $5.161,75 dólares.

* 1. **COSTO ESFUERZO HOMBRE/MES CON MONTE CARLO:**

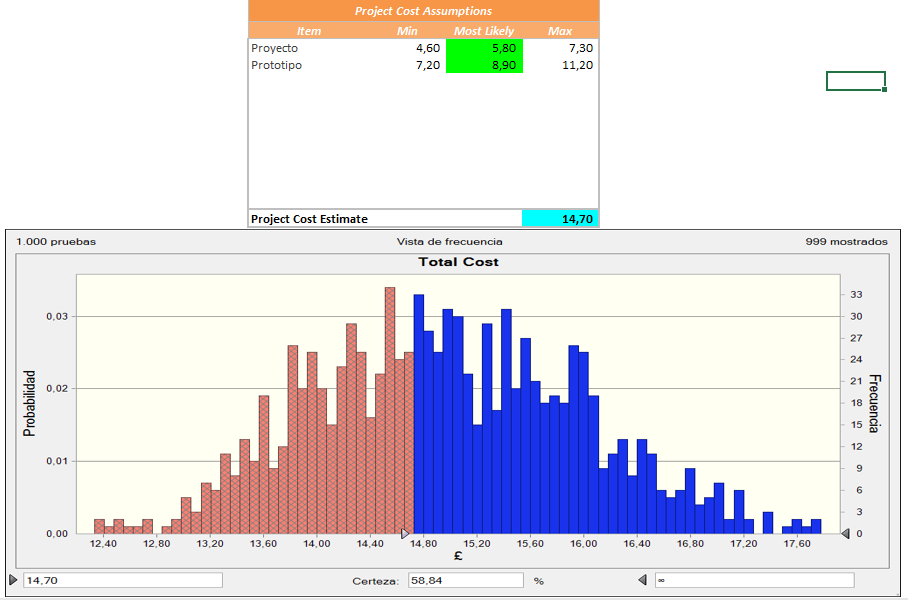


Imagen 6: Imagen que permite visualizar la estimación por costo o esfuerzo Hombre/mes con Monte Carlo

De esta manera podemos concluir que nuestro proyecto tendrá un esfuerzo hombre/mensual total estimado en 14,70.

1. **APLICANDO TDD:**

La estrategia o Test Driven Development por sus siglas en ingles TDD es una práctica de ingeniería de software que consiste en unir dos prácticas. Escribir las pruebas primero (Test First Development) y Refactorización (Refactoring). (TDD: Test Driven Development, 2007).

Para el desarrollo del proyecto “Implementación de un prototipo basado en realidad virtual para la enseñanza de cálculo multivariado y vectorial para estudiantes de Ingeniería de Sistemas y Computación de la Universidad del Quindío”, se aplicará la estrategia TDD el cual consiste en seguir tres pasos claves:

1. Realizar un Test automático o prueba unitaria, ejecutarlo y ver que falle. (Test)
2. Hacer el código mínimo para pasar la prueba (FailsTest)
3. Arreglar el código y comprobar que la prueba pasa. (Refactoring)

Para la estimación con la estrategia TDD se tiene en cuenta la estimación inicial del proyecto propuesto en la tabla. Tabla 1: Estimación de cada actividad o driver, la cual ya tiene un porcentaje para la realización de cada actividad.

En la siguiente tabla encontrará la estimación de driver bajo la estrategia TDD.

Tabla 5: Estimación de driver bajo la estrategia TDD.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Driver** | **Estimación Inicial** | **ACTIVIDADES TDD** | | | **Total TDD** | **Estimación Total** |
| **Test** | **Test** | **Refactoring** |
| **D** | Diseño interfaz | 14% | 0 | 0 | 0 | 0 | 14% |
| **E** | Modelos 3D | 20% | 0 | 0 | 0 | 0 | 20% |
| **F** | Escenarios | 14% | 0 | 0 | 0 | 0 | 14% |
| **G** | Integrar modelos y escenarios | 34% | 0 | 0 | 0 | 0 | 34% |
| **H** | Animaciones | 15% | 0.30% | 0.5% | 40% | 41% | 55.80% |
| **I** | Scripts de control | 28% | 0.3% | 0.5% | 40% | 41% | 68.80% |
| **J** | Compilar y crear APK | 21% | 0.2% | 0.05% | 0.05% | 0% | 21.25% |
|  | **Total** | **146%** |  |  |  |  | **227.85%** |

Es importante destacar que el proyecto “Implementación de un prototipo basado en realidad virtual para la enseñanza de cálculo multivariado y vectorial para estudiantes de Ingeniería de Sistemas y Computación de la Universidad del Quindío”, se realizará bajo el ambiente de trabajo Unity el cuál se centra en ambiente gráfico cómo sus requisitos lo mencionan en la Tabla 5: Estimación de driver bajo la estrategia TDD. , y el TDD no es aplicable para crear test sobre la interfaz (Herranz Roldán, 2011), por lo tanto no es colaborativo para el diseño de la misma. Es por ello que en la estimación que muestra la tabla anterior, Tabla 4: Estimación de cada actividad o driver., los nodos (drivers) D, E, F, G no tienen valor asignado ya que corresponde a requisitos propios del diseño y desarrollo de la interfaz, mientras que para los nodos H, I, J, animaciones, Scripts de control y Compilar y crear APK respectivamente si tienen implementación TDD. Lo cual aumenta considerablemente la estimación de cada uno de los ítems, como se puede observar en la siguiente Tabla 6: Diferencia de estimación inicial respecto a estimación TDD:

Tabla 6: Diferencia de estimación inicial respecto a estimación TDD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Driver** | **Estimación Total** | **Estimación Inicial** | **Diferencia** |
|
| **H** | Animaciones | 55.80% | 15% | 40.80% |
| **I** | Scripts de control | 68.80% | 28% | 40.80% |
| **J** | Compilar y crear APK | 21.25% | 21% | 0.25% |
| **Total** | | **145.85%** | **64%** | **81.85%** |

Adicional a las pruebas unitarias implementadas para el desarrollo, cabe resaltar que los factores de cambio mencionados en la Tabla 1: Identificación de factores de cambio. Pueden cambiar la estimación del proyecto, es por ello que se debe hacer la respectiva revisión del proceso durante la entrega de cada sprint.

# Bibliografía

Herranz Roldán, J. I. (17 de Enero de 2011). *Tecnología para desarrollo.* Obtenido de Tecnología para desarrollo: https://www.paradigmadigital.com/dev/tdd-como-metodologia-de-diseno-de-software/

*TDD: Test Driven Development.* (2007). Obtenido de TDD: Test Driven Development: http://www.chuidiang.com/java/herramientas/test-automaticos/tdd-test-driven-development.php