

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA,
INFORMÁTICA Y MECÁNICA
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y SISTEMAS



SEGUNDA ENTREGA: AVANCE SISTEMA DE TUTORÍAS

ASIGNATURA: Ingeniería de Software I

DOCENTE: Quintanilla Portugal, Roxana Lisette

ALUMNOS:

- | | |
|-----------------------------------|--------|
| • Bustamante Flores, Erick Andrew | 171943 |
| • Choque Sarmiento, Leydi Diana | 174909 |
| • Huancara Ccolqqe, Alex Helder | 174911 |
| • Inca Cruz, Carlos Eduardo | 174912 |
| • Luycho Ancaifuro, Mariela | 170436 |
| • Sarco Jacinto, Daniel Eduardo | 174452 |
| • Tito Saya, Alexander | 164249 |

SEMESTRE 2021-I

CUSCO – PERÚ

2021

Contenido

USO DE HERRAMIENTAS PARA DESARROLLAR EL SISTEMA DE TUTORIAS.....	4
• Azure DevOps:.....	4
• Microsoft Visual Studio (IDE):	4
• Lenguaje de programación C#:.....	4
• Management Studio:	4
• SQL Server:	4
• GitHub:	4
BACKLOG	5
WORK ITEMS.....	5
SPRINT	7
ANÁLISIS DE COSTO DEL SOFTWARE	8
COCOMO.....	8
Modo de desarrollo: Modo Orgánico (Organic).....	8
ANÁLISIS DE RIESGO	12
Requisitos.....	12
Estabilidad / integridad	12
Claridad	12
Viabilidad.....	12
Seguimiento.....	12
Diseño	12
Funcionalidad	12
Dificultad.....	13
Interfaces.....	13
Rendimiento y calidad	13
Capacidad de prueba	13
Restricciones de hardware.....	14
Reutilización de software.....	14
Prueba de código y unidad	14
Viabilidad.....	14
Pruebas.....	14
Codificación / Implementación	14
Integración y Prueba.....	15
Entorno.....	15
Producto.....	15
Sistema	15

Mantenibilidad	16
Especificaciones	16
Comunicación, compatibilidad de equipo y motivación	16
Comunicación	16
Compatibilidad del equipo	16
Motivación del equipo	16
ANÁLISIS GENERAL DE RIESGOS EN GRAFICOS	17
ANÁLISIS GENERAL	17
ANÁLISIS POR CATEGORÍA	17

USO DE HERRAMIENTAS PARA DESARROLLAR EL SISTEMA DE TUTORIAS

- **Azure DevOps:** Usado para administrar nuestro avance del proyecto, para poder realizar el trabajo de mejor forma, con mayor calidad, tiene versión gratuita.
- **Microsoft Visual Studio (IDE):** IDE Entorno de desarrollo propio de Microsoft. Usado por la integración fácil integración de SQL Server y C# para desarrollo de proyectos.
- **Lenguaje de programación C#:** Utilizada para la implementación de código fuente. Lenguaje orientado a objetos, es un lenguaje fácil de usar, los integrantes de equipo de desarrollo conocen el lenguaje y tiene experiencia en el desarrollo de sistemas.
- **Management Studio:** Gestor de base de datos para manejar de forma rápida la base de datos.
- **SQL Server:** Implementación de la base de datos.
- **GitHub:** Repositorio donde almacenaremos los avances del proyecto en la nube.

Herramientas



Azure
DeVops



Visual
Studio + C Sharp



GitHub


































BACKLOG

La implementación del Backlog para el Sistema de Tutorías se realizó en la plataforma de Azure DevOps

WORK ITEMS

ID	Title	Assigned To	State	Area
23	✓ 07.1.Implementar procedimientos almacenados asignar estudiante...	Unassigned	To Do	Sist
22	✓ 04.3.Implementar funcionalidad asignar y filtrar tutores a grupo de...	Unassigned	To Do	Sist
<input type="radio"/> 21	✓ 04.2.Implementar interfaz asignar y filtrar tutores a grupo de ...	Unassigned	To Do	Sist
20	✓ 04.1.Implementar procedimientos almacenados asignar y filtrar tut...	Unassigned	To Do	Sist
12	✓ 03.3.Implementar interfaz del Login	Unassigned	To Do	Sist
13	✓ 03.2.Implementar funcionalidad del Login	Unassigned	To Do	Sist
11	✓ 03.1.Implementar procedimientos almacenados del Login	Unassigned	To Do	Sist
9	✓ 02.2.Mantenimiento de tablas(CRUDS)	Unassigned	To Do	Sist
4	✓ 01.3.Modelado SD	Unassigned	To Do	Sist
54	👑 06. Asignar y filtrar tutores a Grupo de estudiantes	Unassigned	To Do	Sist

ID	Title		Assigned To	State	Area
 64	 01. Planificación del Sistema	...	 Unassigned	 To Do	Siste
10	 01.2.Modelado del funcionamiento del sistema		 Unassigned	 To Do	Siste
42	 13.3.Solución de errores encontrados en el sistema		 Unassigned	 Doing	Siste
57	 09. Asignar ayudantes de tutoria a Aulas de Tutoria		 Unassigned	 Doing	Siste
5	 01.4Modelado SR		 Unassigned	 To Do	Siste
61	 13. Pruebas y validacion		 Unassigned	 To Do	Siste
60	 12. Registrar Informe Semestral de Tutoria		 Unassigned	 To Do	Siste
59	 11. Registrar Fichas de Tutoria		 Unassigned	 To Do	Siste
58	 10. Control de Asistencia a Aulas de Tutoria		 Unassigned	 To Do	Siste
56	 08. Asignar horarios a Aulas de Tutoria		 Unassigned	 To Do	Siste

SPRINT

Taskboard Backlog Analytics

+ New Work Item ...

🔄 Sprint 1 ▾

Person: All




^ Collapse all

■ Unparented


To Do

✓ **10** 01.2.Modelado del funcionamiento del sistema

 Unassigned


State ● To Do

11 03.1.Implementar procedimientos almacenados del Login

 Unassigned

State ● To Do

12 03.3.Implementar interfaz del Login

 Unassigned


State ● To Do

13 03.2.Implementar funcionalidad del Login

 Unassigned

Doing

9 02.2.Mantenimiento de ...
tablas(CRUDS)

 Unassigned

State ● Doing

✓ **20** 04.1.Implementar procedimientos almacenados asignar y filtrar tutores a

 Unassigned

State ● Doing


✓ **21** 04.2.Implementar interfaz asignar y filtrar tutores a grupo de estudiantes

 Unassigned

State ● Doing


Done

2 Captura de Requisitos

 Unassigned


State ● Done

3 Filtrar los requisitos del alta prioridad

 Unassigned


State ● Done

5 01.4Modelado SR

 Unassigned

State ● Done

✓ 6 01.5.Crear prototipos de la interfaz del sistema

 Unassigned

State ● Done

ANÁLISIS DE COSTO DEL SOFTWARE

Para la estimación del costo del proyecto se hizo en COCOMO, Se usó como base la versión de Barry W. Boehm, se hizo respectivas mejoras de acuerdo a los requerimientos necesarios para el desarrollo del software.

COCOMO

Modo de desarrollo: Modo Orgánico (Organic)

Modelo básico

Modelo básico				
Estima el esfuerzo y el tiempo empleado en el desarrollo de un proyecto de software usando dos variables predictivas denominadas factores de costo (cost drivers): el tamaño del software y el modo de desarrollo.				
$LDC = 120 * FDE/S$	Donde: $FDE/S = \text{Flujo de salida} + \text{Flujo de entrada} = PF * PF$			PF: Metricas de puntos de funcion
Flujo de entrada=	10		Flujo de salida=	10
$FDE/S =$	20			
$LCD =$	4556			
Lineas de codigo(KSLOC) $\Rightarrow KSLOC = LDC/1000$	Coeficiente segun modo de desarrollo(A)	Coeficiente Segun modo de desarrollo(B)	Coeficiente segun modo de desarrollo(C)	Coeficiente Segun modo de desarrollo(D)
4,556	2,4	1,05	2,5	0,38

Esfuerzo:

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

Cronograma:

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

Modo de desarrollo	Esfuerzo (PM)-Meses persona	Cronograma (TDEV)-Meses de trabajo
Organico	11,80	6,39
Costo Hombre Maquina \Rightarrow CHM=	1500	
Estimación del personal necesario (Personas) CP=PM/TDEV	Estimación de productividad (Velocidad) P=LDC/PM	Estimación del costo (Soles) C=CHM*PM
1,85	386,2420669	17693,56729
2 personas	399 Instrucciones_persona_mes	Costo del proyecto (Soles)
Observaciones:		
Un mes-persona equivale a 152 horas de trabajo y corresponde a la cantidad de tiempo que una persona dedica durante un mes a trabajar en un proyecto de desarrollo de software. Este valor tiene en cuenta los fines de semana, pero excluye feriados y vacaciones.		
KSLOC es una unidad de medida, donde 1 KSLOC equivale a mil líneas de código fuente		

Modelo intermedio

Este modelo incorpora un conjunto de quince variables de predicción que toman en cuenta las variaciones de costos no consideradas por COCOMO Básico.

Los factores seleccionados se agrupan en cuatro categorías:

1. Atributos del producto de software

2. **Atributos del hardware**
3. **Atributos del personal involucrado en el proyecto**
4. **Atributos propios del proyecto**
 - a. Cálculo del Esfuerzo Nominal
LCD = 4556

Líneas de código (KSLOC)⇒KSLOC=LDC/1000	Coeficiente según modo de desarrollo(A)	Coeficiente Según modo de desarrollo(B)	Coeficiente según modo de desarrollo(C)	Coeficiente Según modo de desarrollo(D)
4,556	3,2	1,05	2,5	0,38

	PM= A*(KSLOC)^B
PM =	15,72761536

- b. **Determinar el Factor de Ajuste del Esfuerzo (EAF)**

Factores	Factor Multiplicador
RELY	0,88
DATA	0,94
CPLX	0,85
TIME	1
STOR	1
VIRT	1
TURN	1
ACAP	1

AEXP	0,91
PCAP	0,86
VEXP	0,9
LEXP	1
MODP	1
TOOL	0,91
SCED	1,04
EAF=	0,4686909158

c. Esfuerzo Ajustado

	PM= $A * EAF * (KSLOC)^B$
PM =	7,37

d. Cronograma

	$TDEV = C * (PM)^D$
TDEV=	5,340830395

Estimación del personal necesario (Personas) $CP = PM / TDEV$	Estimación de productividad (Velocidad) $P = LDC / PM$	Estimación del costo (Soles) $C = CHM * PM$
1,38	618,0652119	11057,08567
2 personas	276 Instrucciones persona_mes	Costo del proyecto (Soles)

ANÁLISIS DE RIESGO

Requisitos

Estabilidad / integridad [evaluado mediante la evaluación de la cantidad de información en los requisitos]

[1] ¿Están cambiando los requisitos o aún no se han determinado? **SI**

- Lo más recomendable es que los requisitos sean totalmente definidos al inicio del proyecto, pero ya que aún estamos en un proceso de aprendizaje muchas veces a medida que los vamos definiendo obviamos algunos y otras veces modificamos otros ya que por algún motivo no fueron identificados al inicio o no fueron definidos de la manera correcta, esto podría llevar a que, al momento de modificarlos, aquellos que dependan de él vayan a también ser modificados.

[2] ¿Tiene el instructor requisitos / expectativas no escritas? **NO**

- No, ya que los requisitos fueron identificados desde un documento el cual fue asignado por la docente.

Claridad [evaluada mediante la evaluación de su comprensión de los requisitos]

[3] ¿Puede comprender los requisitos tal como están escritos? **NO**

- No, en algunos casos los requisitos extraídos no son bien definidos ya que no están descritos de una manera entendible para el desarrollador.

Viabilidad [evaluada mediante la evaluación de las posibles dificultades que puedan surgir más adelante en el proyecto]

[4] ¿Existen requisitos que sean técnicamente difíciles de implementar? **SI**

- Si, muchas veces hay requisitos que no tienen una definición correcta lo cual no nos permite reconocer la implementación de este.

Seguimiento [evaluado mediante la evaluación de la capacidad de mantener visibles los requisitos durante el proyecto]

[5] ¿Planea hacer el seguimiento de la identificación de requisitos de manera proactiva o reactiva? **REACTIVA**

- De manera reactiva ya que al no tener mucha experiencia desarrollando software muchas veces no nos damos cuentas de los errores al inicio y ya en el momento que sucede.

[6] ¿Tiene un plan para realizar un seguimiento de los requisitos a lo largo de las fases de diseño, codificación y prueba? **SI**

- El plan será usar el modelo de manera reactiva el cual nos permitirá resolver los errores en el momento que aparezca esto quitará tiempo, pero se hará en el momento oportuno.

Diseño

Funcionalidad [evaluada mediante la evaluación del conjunto de características y capacidades del producto]

[7] ¿Existe algún algoritmo específico que no satisfaga (o sólo parcialmente) los requisitos? **SI**

- En el proceso de desarrollo esperamos no encontrarnos con este tipo de problemas, pero por la poca experiencia muchas veces definimos procesos los cuales en nuestra mente se nos hace fácil o creemos que cubrirá de manera correcta el proceso, pero al llegar el momento de desarrollarlo tenemos que separar el proceso lo cual nos llevará a evaluarlo de manera distinta.

[8] ¿El tiempo en la implementación de algoritmos será el correcto? **NO**

- No, la poca experiencia en el ámbito de desarrollo de software no permitirá que podamos ver de manera concisa el tiempo de este proceso.

[9] ¿Existen algoritmos que puedas reutilizar facilitando el proceso de desarrollo? **SI**

- Si, en internet y en librerías de las cuales podríamos importar algoritmos los cuales facilitarían el desarrollo de este software.

Dificultad [evaluada mediante la evaluación del esfuerzo involucrado en la producción del diseño]

[10] ¿Algo del diseño depende de suposiciones optimistas o poco realistas? **POCO REALISTAS**

- Si, muchas veces nos hacemos la idea de que todo puede funcionar cumpliendo lo requerido, pero al momento de desarrollarlo no funciona o debe ser separado, definiendo así poco realista el diseño.

[11] ¿Existen requisitos o funciones que sean difíciles de diseñar? **SI**

- Si, existen algoritmos complejos que muchas veces deben ser reevaluados, así como dicho requisito, esto ya que algunos son poco realistas.

Interfaces [evaluadas mediante la evaluación de las conexiones entre componentes o con el mundo exterior]

[12] ¿Están bien definidas las interfaces internas y externas? **NO**

- No, muchas veces las interfaces se van modificando a medida que el proyecto avanza para así poder cumplir con el sistema.

Rendimiento y calidad [evaluados mediante la evaluación de la funcionalidad y la calidad del producto]

[13] ¿Existe algún problema con el rendimiento o la calidad esperados del diseño? **SI**

- Para el grupo el problema estaría en el rendimiento, consideramos que el rendimiento puede llegar hasta a un 60% - 80% ya que algunos módulos quizá no puedan ser concluidos.

[14] ¿Existe algún problema en la estética y en la apariencia del software? **SI**

- En el tiempo de desarrollo se debe cubrir todos estos factores, pero muchas veces al estancarnos en el desarrollo del código o en el proceso de identificación de requerimientos no se llega a dar la correcta estética y apariencia al software.

Capacidad de prueba [evaluada mediante la evaluación del esfuerzo requerido para probar suficientemente el producto]

[15] ¿Será fácil probar el software? **SI**

- Una vez concluidos todos los procesos, el inicio del proceso de pruebas seria uno de los más importantes ya que este nos permitirá saber si existe o no problemas dentro del software, este software al tener una base de datos pequeña y procesos no muy complejos, sería fácil de probarlo.

Restricciones de hardware [evaluadas mediante la evaluación del hardware de la plataforma de desarrollo o de destino]

[16] ¿El desarrollo o el hardware de destino limitan su capacidad para cumplir con los requisitos? **SI**

- Primero tendríamos que identificar los usuarios finales que en este caso serian docente y alumnos.

El software no tendrá una complejidad grande, no requiere de computadoras de gama alta, y al ser docentes y estudiantes de la Carrera de Ing. Informática y de Sistemas estos contarán con equipos los cuales permitirá que el usuario final pueda usarlo de manera correcta.

Reutilización de software [evaluado mediante la evaluación de la medida en que se reutiliza el software en el producto]

[17] ¿Existe software reutilizado o rediseñado? **SI**

- Muchas veces el software reutilizado al modificarlo genera más problemas que al volver a programarlo por ellos es mejor determinar esto antes.
- Nuestro software tiene algunos procesos iguales lo cuales podemos reutilizarlos al no tener un nivel de complejidad altos estos al momento de ser modificados no generarán problemas.

Prueba de código y unidad

Viabilidad [evaluada mediante la evaluación de la relativa facilidad necesaria para realizar el código y la prueba]

[18] ¿Los algoritmos podrán tener un seguimiento a lo largo de vida del sistema? **SI**

- El equipo de desarrollo reconoce que puede haber mejoras lo cual permitirá que estos puedan tener un seguimiento y ser modificados a futuro.

Pruebas

[19] ¿Hay tiempo suficiente para realizar todas las pruebas unitarias que especificó? **SI**

- Si ya que lo primordial es poder cumplir los requisitos y las pruebas unitarias nos permitirán un análisis completo de estos.

Codificación / Implementación

[20] ¿Las especificaciones de diseño son lo suficientemente detalladas para escribir el código? **NO**

- No, las especificaciones en el diseño muchas veces varían ya que a veces se encuentran errores en el código y esto genera pérdidas de tiempo.

[21] ¿Está cambiando el diseño mientras se realiza la codificación? **No**

- Aún no nos encontramos en etapa de codificación, pero posiblemente haya un cambio mínimo

[22] ¿Es el lenguaje adecuado para producir el software de este programa? **NO**

- No, por la limitación de tiempo para el desarrollo, estamos optando un lenguaje conocido para escritorio lo cual no es el adecuado

[23] ¿Su equipo tiene suficiente experiencia con el lenguaje, la plataforma o las herramientas de desarrollo? **SI**

- El equipo ya tiene experiencia en desarrollo de escritorio en los lenguajes y plataformas

[24] ¿Existe el riesgo de que un componente o módulo clave no esté completo o no esté programado? **Si**

- Siempre existe la posibilidad, pero pensamos solucionarlo en el momento

[25] ¿Se siente cómodo con la estimación de su equipo sobre el tiempo y el esfuerzo de codificación? **No**

- Aún no se sabe porque no se inició con la codificación

[26] ¿Tiene un plan para la gestión de la configuración del código? **No**

- Trabajar en una aplicación para escritorio es muy difícil, ya que puede haber alguna modificación de código ya que la plataforma de trabajo no es compartida en tiempo real en la nube

Integración y Prueba

Entorno [evaluado mediante la evaluación de las instalaciones de soporte de hardware y software y los casos de prueba]

[27] ¿Habría suficiente hardware para realizar la integración y las pruebas adecuadas? **No**

- Limitación en equipos de hardware
- Falta de tiempo para culminar el proyecto

[28] ¿Existe algún problema con el desarrollo de escenarios realistas y datos de prueba para demostrar los requisitos? **Si**

- No cumplir con la culminación del proyecto por falta de tiempo por consiguiente también las pruebas

Producto [evaluado mediante la evaluación de la integración y prueba de grupos de componentes]

[29] ¿Se han acordado criterios de aceptación para todos los requisitos? **Si**

- Acorde al avance del sistema, inclusión de nuevos requisitos necesarios

[30] ¿Se ha especificado suficiente integración de productos y se ha asignado el tiempo adecuado para ello? **No**

- Poco tiempo para el desarrollo del proyecto.

Sistema [evaluado mediante la evaluación de la integración entre el producto y el hardware de destino]

[31] ¿Se ha especificado suficiente tiempo de integración del sistema? **No**

- No cumplir con el cronograma o integración apresurada del sistema

Mantenibilidad [evaluada mediante la evaluación del esfuerzo requerido para localizar y corregir errores]

[32] ¿El diseño y la documentación del producto son adecuados para que otra clase mantenga el código? **Si**

- Prototipo legible y documentación entendible

Especificaciones

[33] ¿Son las especificaciones de prueba adecuadas para probar completamente el sistema? **No**

- No tenemos una base sólida de requisitos porque son cambiantes, por consiguiente, no inclusión de nuevas especificaciones.

Comunicación, compatibilidad de equipo y motivación

Comunicación [evaluada mediante la evaluación de la capacidad del equipo para intercambiar información]

[34] ¿Existe una falta de buena comunicación entre su equipo? **Si**

- Carga académica de los miembros del equipo
- Horarios no disponibles para reuniones o cruce de horarios de clases y reuniones

[35] ¿Hay una falta de buena comunicación con su instructor sobre el proyecto? **Regular**

- La información que imparte la instructora es imprecisa
- Información de distintas fuentes genera Confusión

Compatibilidad del equipo [evaluada mediante la evaluación de la capacidad del equipo para trabajar de manera productiva]

[36] ¿Conoce su equipo? **Regular** ¿Han trabajado juntos en un proyecto de equipo antes? **Regular**

- Formación de un nuevo equipo de desarrollo (Trabajos parciales realizados con algunos integrantes del equipo).
- Falta de rápida adaptación a un nuevo equipo

[37] ¿Se delegan las tareas de manera justa entre su equipo? **Si**

- Trabajos delegados de forma democrática mediante sorteos

Motivación del equipo [evaluado mediante la evaluación de los objetivos del equipo]

[38] ¿Su equipo está motivado para crear un buen producto? **No**

- Motivación sólo calificaciones (motivos externos)
- Carga académica excesiva por parte de los integrantes del equipo
- Tiempo insuficiente para trabajos de investigación constante

ANÁLISIS GENERAL DE RIESGOS EN GRAFICOS

ANÁLISIS GENERAL

	Identificación de riesgo	
Proyecto	Riesgo	No Riesgo
Software	25	13

ANÁLISIS POR CATEGORÍA

	Identificación de riesgo	
Categorías	Riesgo	No Riesgo
Requisitos	4	2
Diseño	7	4
Prueba de código y unidad	5	4
Integración y prueba	5	2
Comunicación, compatibilidad de equipo y motivación	4	1
TOTAL	25	13

