**Guía de Proyecto:**

**Proyecto de Software con Aplicación de Metodologías Ágiles**

**Descripción de la Asignatura y Plan de trabajo.**

La asignatura de Taller de Diseño y desarrollo de Soluciones es una asignatura Hito de carácter práctica, dictada en modalidad presencial, del área formativa de la Especialidad de Tecnologías de Información y Ciberseguridad. Mediante el progreso de este taller, el estudiante demostrará el avance de sus competencias de egreso en escenarios de desempeño próximas a la realidad laboral, a través de diseño y desarrollo de aplicaciones de software integradas que incluyan tecnologías e infraestructuras emergentes.

El propósito de la asignatura es asegurar el desarrollo progresivo de las competencias de los estudiantes por medio de la estrategia metodológica Aprendizaje Basado en Proyectos (ABPro) y de actividades de retroalimentación.

El plan de trabajo consta de cuatro etapas que abarcan las dos unidades establecidas en el programa de asignatura. Cada etapa representa la progresión de un proyecto, con sus respectivos productos entregables y plazos, los cuales se detallan a continuación:

Tabla 1. Plan de trabajo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Unidad** | **Etapa** | **Avances esperados y actividades** | **Avances y entregas del proyecto (Fecha aproximada)** |
| 1 | Levantamiento de requerimientos y diseño de interfaces. | * Desarrollo de historias de usuarios. * Desarrollo de casos de uso de los primeros sprint. * Diseño de las interfaces de los primeros sprint. | * Informe de requerimientos. * Diagramas de casos de uso. * Interfaces de usuario.   (Semana 4). |
| 1 | Diseño de la solución. | * Diseñan solución aplicando patrones de arquitectura y almacenamiento en cloud. * Desarrollan el plan de pruebas. | * Patrones de arquitectura. * Almacenamiento en cloud. * Plan de pruebas.   (Semana 6). |
| 2 | Construcción de la solución. | * Desarrollo de la aplicación incluyendo interfaces, codificación, implementación de la base de datos en cloud e instalación. * Configuran el entorno de la aplicación en producción. | * Trabajo práctico incluyendo interfaces, codificación, implementación de la base de datos en cloud e instalación. * Configuración del entorno de producción.   (Semana 11). |
| 2 | Defensa de la solución en exposición de resultados. | * Desarrollan interfaces, codificación. * Plan de pruebas. | * Trabajo práctico. * Código fuente de la aplicación. * Ejecutable de la aplicación. * Documentación de la aplicación.   (Semana 17-18) |

Fuente: Elaboración Propia (2021).

**Organización de los equipos de trabajo.**

Las actividades durante el semestre se realizarán en equipos de trabajo para fortalecer las habilidades de organización y trabajo en equipo de los estudiantes. Por lo cual, se recomienda que la distribución de los equipos sea realizada por el docente, considerando un máximo de 3 integrantes por grupos de trabajo, definiendo junto con los estudiantes los roles y responsabilidades dentro del equipo.

**Unidad 1: Evaluación Sumativa 1.**

**Levantamiento de requerimientos y diseño de interfaces**

1. Presentación

La siguiente guía de proyecto entrega las orientaciones para la realización del proyecto de software con aplicación de las metodologías ágiles, integrando la base de datos y la programación orientada a objetos.

En esta primera etapa los equipos de trabajo deberán identificar las necesidades del negocio utilizando distintos instrumentos para el levantamiento de los requerimientos y confeccionando las historias de usuario para el posterior diseño de los respectivos casos de uso de los primeros sprint.

La evidencia de la evaluación sumativa 1 será un informe sobre los requerimientos, los diagramas de casos de uso de los primeros sprint, y las respectivas interfaces de usuario.

* **Criterios de evaluación**

1.1.1.- Efectúa el levantamiento de los requerimientos utilizando instrumentos acordes al perfil del cliente.

1.1.2.- Genera los sprint en base a las historias de usuario.

1.1.3.- Considera aplicación de paradigma 4 + 1.

1.1.4.- Diseña interfaces gráficas modernas (intuitiva, minimalista, simple).

**Unidad de Aprendizaje 1:**

Diseño de la Solución.

**Aprendizaje esperado**

En esta situación de desempeño, el estudiante diseñará una solución de software que responde a las necesidades del negocio. (Integra Competencia genérica trabajo en Equipo, Nivel 1).

**Actividades**

1. Identificar las necesidades del negocio mediante el levantamiento de los requerimientos.
2. Priorizar las historias de usuario y generar los primeros sprint.
3. Diseñar los diagramas UML según las distintas vistas de acuerdo con el paradigma 4 + 1.
4. Diseñar las interfaces.

**Evaluación**

* Evaluación sumativa con escala de apreciación/pauta de corrección del informe.
* Ponderación 15%

1. Instrucciones
2. Conformar grupos de trabajo con un máximo de 3 estudiantes. Los grupos o equipos de trabajo podrán ser organizados por el docente, atendiendo a la realidad laboral.
3. Una vez conformados los grupos de trabajo, seleccionan al estudiante que deberá asumir el rol de jefe de proyecto, quien será el representante del equipo, encargado de relacionarse con el docente y el responsable de cumplir con las instrucciones y actividades a desarrollar. Este rol debe ir rotando durante el transcurso del proyecto, de tal forma que cada integrante tenga la oportunidad de asumir diferentes responsabilidades.
4. Establecer un proyecto informático que solucione un problema, preferentemente de la vida real.
5. Actividades

Para el desarrollo de esta primera etapa del proyecto, se deberán realizar cuatro actividades:

1. Definir el proyecto a resolver y efectuar el levantamiento de los requerimientos utilizando los instrumentos adecuados.
2. Priorizar las historias de usuario a partir de los requerimientos levantados y generar los primeros sprint.
3. Aplicar paradigma 4 + 1 para representar las distintas vistas y confeccionar los casos de uso a partir de los requerimientos anteriores.
4. Confeccionar las interfaces de usuario y que responden a los casos de uso.
5. **Definir el proyecto a resolver y efectuar el levantamiento de los requerimientos utilizando los instrumentos adecuados.**
6. El diagnóstico de la situación tiene como propósito conocer algún problema que tenga un posible cliente (persona natural o empresa) y que requiera de una solución tecnológica. Estas ideas de negocio pueden tratarse de un emprendimiento, es decir, una empresa que desea poner un nuevo producto en el mercado; o bien, una innovación tecnológica, es decir, implementar una solución informática en alguna empresa que desea mejorar algunos de sus procesos dentro su cadena de valor.
7. El levantamiento de los requerimientos deberá escribirse en la forma de historias de usuario, señalando el rol, la funcionalidad que se requiere, la razón y los criterios de aceptación.

Por ejemplo, si el usuario nos presenta el siguiente requerimiento no funcional:

RNF01: La imagen de la página web que será desarrollada debe mantener la imagen corporativa.

La historia de usuario es la siguiente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RECOPILACIÓN DE HISTORIAS DE USUARIO** | | | | |
| **ID** | **Rol** | **Característica / Funcionalidad** | **Razón / Resultado** | **Criterios de aceptación** |
| HU01 | Como Usuario de la página web | Necesito visualizar que todos los componentes de la página se identifiquen con la imagen corporativa | Con la finalidad de sentirme identificado con la Institución | * Los botones deben ser circulares. * Los colores predominantes deben ser el rojo, amarillo y blanco según panteón adjunto. * El logo de la institución debe visualizarse siempre al lado superior derecho. |

1. **Priorizar las historias de usuario y generar los primeros sprint**.

Es responsabilidad del *product owner* (dueño de producto) priorizar las historias de usuario considerando el valor de negocio.

1. El grupo de trabajo, en común acuerdo debe definir el tiempo asignado al sprint. Usualmente el largo del sprint permanece constante durante todo el proyecto y puede ir de 1 a 4 semanas.
2. Definir las historias junto con sus respectivas actividades que ingresarán a los primeros sprint. Cabe señalar que las primeras en ingresar serán aquellas con más alta prioridad; sin embargo, si se detecta que existe alguna historia de baja prioridad que esté relacionada con una de alta, entonces, debe incluirse previo a generar sus actividades y cuantificar su tiempo.
3. La suma de los tiempos de cada actividad no debe sobrepasar el tiempo asignado al sprint. Si eso ocurriera, dicha historia debe programarse para el sprint siguiente.
4. Definir las entregas al usuario. Estas pueden ser al término de 1 o más sprint. Cada entrega debe proporcionar valor significativo al usuario.
5. **Diseñar los diagramas UML** **según las distintas vistas de acuerdo con el paradigma 4 + 1.**
6. Diseñar la vista lógica mediante el diseño del diagrama de clases y de secuencia.
7. Diseñar la vista de desarrollo mediante el diagrama de componentes.
8. Diseñar la vista física mediante el diagrama de despliegue.
9. Diseñar la vista de procesos mediante el diagrama de actividades.
10. Diseñar los casos de uso de las historias que entrarán en los primeros sprint.
11. **Confeccionar las interfaces de usuario y que responden a los casos de uso**.

Construir las interfaces de usuario y que respondan a los casos de uso de las historias de usuario que han sido consideras en los primeros sprint y que correspondan al primer entregable de valor.

1. Recursos e instrumentos de evaluación del aprendizaje

| Recursos de apoyo para las actividades | Instrumento(s) de evaluación |
| --- | --- |
| Plantilla: Guía Informe de Proyecto Evaluación sumativa 1 | Rúbrica 1: Informe Formulación de Proyecto |

**Evaluación Sumativa 2.** **Arquitectura de la solución, almacenamiento y plan de pruebas**

1. Presentación

En esta segunda etapa los equipos de trabajo definen patrones de arquitectura, incluyen en el diseño los estándares de programación segura, el almacenamiento de los datos en cloud, y diseñan el plan de pruebas.

La evidencia de la evaluación sumativa 2 será un informe que detalle y justifique los patrones de arquitectura a utilizar y el almacenamiento en cloud, junto con el detalle del plan de pruebas.

* **Criterios de evaluación**

1.1.5.- Define patrones de arquitectura a utilizar (MVC).

1.1.6.- Incluye en el diseño los estándares de tecnología y programación segura.

1.1.7.- Considera almacenamiento en plataforma cloud.

1.1.8.- Diseña plan de pruebas.

1.1.9.- Cumpliendo con las tareas asignadas en el tiempo definido.

**Unidad de Aprendizaje 1:**

Diseño de la Solución.

**Aprendizaje esperado**

En esta situación de desempeño, el estudiante diseñará una solución de software que responde a las necesidades del negocio. (Integra Competencia genérica trabajo en Equipo, Nivel 1).

**Evaluación**

* Evaluación sumativa con pauta de cotejo.
* Ponderación 25%

**Actividades**

1. Definir los patrones de arquitectura de la solución.
2. Incluir estándares de programación segura.
3. Definir almacenamiento de datos en cloud.
4. Diseñar el plan de pruebas.
5. Instrucciones
6. Continuar con los mismos grupos de trabajo y proyecto conformados en la evaluación anterior.
7. Para esta evaluación se sugiere mantener la asignación de jefe de proyecto en el mismo integrante del equipo anteriormente elegido; rol que irá rotando en las evaluaciones siguientes.
8. Aplicar las correcciones recibidas en la retroalimentación efectuada por el docente al informe de la evaluación anterior, de tal forma de ir mejorando en su proceso formativo.
9. Actividades

Para el desarrollo de esta segunda parte del proyecto, se deberán realizar las siguientes actividades:

1. Definir los patrones de arquitectura de la solución.
2. Incluir estándares de programación segura.
3. Definir almacenamiento de datos en cloud.
4. Diseñar el plan de pruebas.
5. **Definir los patrones de arquitectura de la solución.**
6. Definir el esquema de organización que tendrá la estructura del software. Por ejemplo, MVC, cliente-servidor, n-capas, etc.
7. Comentar con el docente la arquitectura definida a fin de recibir retroalimentación.
8. Incluir en la construcción del sistema la arquitectura definida.
9. **Incluir estándares de programación segura**.
10. Definir perfiles y roles para el sistema.
11. Definir qué privilegios de acceso tendrán los perfiles.
12. Definir y aplicar las buenas prácticas que se implementarán durante el desarrollo; tales como, requisitos de seguridad, validaciones de entrada, simplicidad en el diseño, entre otras.
13. **Definir almacenamiento de datos en cloud.**
14. Investigar los distintos servicios que actualmente existen de almacenamiento, ya sean de pago o gratis, con sus respectivas características.
15. Definir el servicio que más se adecúe según el tipo de proyecto a implementar.
16. Comentar con el docente el servicio escogido.
17. **Diseñar el plan de pruebas.**
18. Identificar los criterios de aceptación de cada historia de usuario.
19. Identificar las funcionalidades que dan solución a las historias de usuario.
20. Definir la estrategia de pruebas, seleccionando los tipos de pruebas a realizar, ya sean funcionales, no funcionales, integrales, de arquitectura, de sistema, entre otras.

IV. Recursos e instrumentos de evaluación del aprendizaje

| Recursos de apoyo para las actividades | Instrumento(s) de evaluación |
| --- | --- |
| Plantilla: Informe Evaluación 2 | Lista de Cotejo 1: Arquitectura de la solución, almacenamiento y plan de pruebas |

**V.** **Bibliografía**

Jiménez M. (2014). *Bases de datos relacionales y modelado de datos (UF1471)*. IC Editorial.

Kaminski S. (2016) Python 3. De Gruyter Oldenburg.

Larman C. (2003). UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado. Pearson Educación.

Pulido E. (2019). *Base de Datos*. Grupo Editorial Patria.

Schwaber K. (2020). La Guía de SCRUM. Disponible en https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf