

INSTRUCCIONES PARA LAS PERSONAS PARTICIPANTES PROYECTO INDIVIDUAL

Valor: 60% (tres etapas de 20% cada una)

Objetivo principal:

Programar aplicaciones informáticas en un entorno orientado a objetos

Objetivos específicos:

1. Crear diagramas de clases UML utilizando entidades, atributos, métodos, herencia, polimorfismo, sobrecarga para definir claramente la estructura del problema.
2. Utilizar el marco de trabajo Scrum para definir prioridades, requerimientos, tareas asociadas, historias de usuario para establecer las necesidades de usuario.
3. Utilizar Microsoft SQL Server Management Studio para definir la estructura de la base de datos.
4. Crear en Visual Studio el modelo de capas para la solución de la aplicación, diseñando los formularios, capas de lógica, capas de acceso a datos y capas de entidades requeridas, para crear una aplicación de escritorio y una aplicación web.

Tipo de evidencias: (Desempeño, Producto, actitud) Trabajo escrito con el diseño de la aplicación y archivo digital con la aplicación programada por la persona participante para dar solución al problema planteado por el docente.

Problema a solucionar

Diseñe y desarrolle una aplicación para administrar el proceso de matrícula de un **Instituto de Idiomas**, tome en cuenta que se ofrece: inglés, francés, alemán y mandarín, cada uno tiene un programa diferente (inglés son 12 cursos con una duración de 80 horas cada uno, francés son 4 cursos con una duración de 100 horas cada uno, alemán son 10 cursos con una duración de 94 horas cada uno y mandarín son 14 cursos con una duración de 120 horas cada uno), las clases son el línea, distribuidas de la siguiente forma: 75% asincrónica, donde el estudiante puede consumir el material en el momento que lo considere pertinente, y el restante 25% debe ser a través de clases sincrónicas, para ello deberá programar la clase con al menos 2 días de anticipación. Cada matrícula es personalizada, por lo que cuando un estudiante se matrícula, el sistema debe indicar la fecha de finalización del curso, para ello el estudiante puede escoger entre 4 niveles de intensidad (bajo = 1 hora por día, medio = 2 horas por día, alto = 3 horas por día, intensivo = 4 horas por día), no debe contabilizar sábados, domingos ni feriados.

El sistema debe realizar como mínimo:

- Los mantenimientos de las tablas (CRUD).
- Matricular un estudiante, debe considerar la cantidad de horas del curso y las horas por día según el nivel seleccionado, debe calcular la fecha de finalización.
- Asignar el horario del profesor, tome en cuenta que el instituto atiende de 8am a 10pm de lunes a viernes, sin embargo, un profesor puede trabajar un máximo de 8 horas por día en el horario que solicite, un profesor solo puede impartir un idioma.
- Agendar horas de estudio sincrónicas, en este caso debe contemplar que la fecha seleccionada sea dos días posteriores a la actual y que se encuentre entre la fecha de inicio y final del curso, además debe contemplar que exista un profesor disponible en el horario solicitado, un profesor solo puede atender un estudiante por clase.
- Si un estudiante no se presenta a la cita sincrónica, pierde las horas de estudio y no se le reprograman, solo puede reprogramar la clase sincrónica con un mínimo de 5 horas de anticipación.
- Debe llevar el control de pagos de los estudiantes, el costo por curso es de \$2 000 por el total de horas del curso, un estudiante no puede matricular si tiene facturas pendientes.

Si usted desea agregar otra funcionalidad al sistema puede hacerlo, sólo debe justificarle al docente la forma en que va a plantearlo.

Fase 1: Diseño de la Aplicación, trabajo escrito (20%)

Descripción general:

Mediante el Lenguaje de Modelado UML diseñe la aplicación solicitada por la persona docente, debe incluir el **diagrama de clases** de la aplicación y la descripción de los requerimientos del sistema mediante **historias de usuario de Scrum**, en este punto debe determinar las tareas necesarias para poder cumplir con cada requerimiento planteado.

Contenido de la fase 1: Debe de presentarse un documento con el siguiente contenido:

1. Portada con sus datos y los datos del módulo
2. Tabla de contenidos (las páginas deben estar numeradas)
3. Introducción
4. Objetivos
5. Problema Empresarial
6. Diseño de la aplicación Orientada a objetos
 - a. Historias de usuario, cada una con:
 - a. Criterios de aceptación
 - b. Tareas asignadas

- c. Reglas de negocio
- b. Requerimientos: describir qué cosas debería hacer la aplicación para que sea una solución funcional, en términos generales.
- c. Prioridades: numerar y describir los procesos que debería hacer la aplicación, empezando por los procesos de mayor prioridad.
- d. Diagrama de clases
- e. Diagrama lógico de la base de datos, tome en cuenta que el diagrama lógico de la BD **se basa en** el diagrama de clases, sin embargo, eso no significa que cada clase del diagrama de Clases necesariamente sea una tabla del diagrama lógico de la BD. Por ejemplo, una **clase abstracta** no es una tabla de la BD.
- f. Diccionario de datos de la base de datos
- 7. Conclusiones
- 8. Anexos (opcionales, si los requiriera)

Fase 2: Aplicación de escritorio (20%)

Descripción: Programación (Código fuente), programación en Visual Studio de la aplicación diseñada en la fase 1, debe incluir el diseño e implementación de la base de datos requerida en SQL Server. En esta fase debe aplicar las correcciones de diseño que sean necesarias, planea la solución con una interfaz para Windows.

Fase 3: Interfaz Web (20%)

Descripción: interfaz web (Código fuente), creación de una interfaz web para la aplicación creada en la fase 2

PROYECTO FASE 1. Valor 20%

Escala numérica de evaluación

Objetivo de la Fase 1: Diseñar aplicaciones informáticas y sus elementos según el paradigma orientado a objetos y los requerimientos de la clientela.

Persona participante:

| Lista de rasgos observables | Criterios | | |
|---|-----------|----|----|
| | CC | CI | II |
| | 2 | 1 | 0 |
| En cuanto a la presentación | | | |
| 1. Incluye portada con todos los datos requeridos | | | |
| 2. Incluye tabla de contenidos y las páginas están numeradas | | | |
| 3. Incluye introducción | | | |
| 4. Incluye objetivos | | | |
| 5. Incluye problema empresarial | | | |
| 6. Incluye conclusiones | | | |
| En cuanto al desarrollo. FASE 1 | | | |
| 7. Crea las historias de usuario utilizando la plantilla correspondiente | | | |

| | | | |
|---|-----------------------------|-----------|--|
| 8. Crea todas las historias de usuario necesarias para definir el funcionamiento del sistema completo | | | |
| 9. Las historias de usuario están narradas de forma correcta y comprensible | | | |
| 10. Las historias de usuario son independientes y describen situaciones de importancia para la definición del sistema | | | |
| 11. Las historias de usuario describen correctamente los criterios de aceptación | | | |
| 12. Las historias de usuario describen correctamente las tareas asignadas | | | |
| 13. Las historias de usuario describen correctamente las reglas del negocio | | | |
| 14. Los requerimientos se detallan adecuadamente para tomar en cuenta todas las tareas | | | |
| 15. Describe adecuadamente las prioridades del sistema explicando cada proceso | | | |
| 16. El diagrama de clases define todas las clases necesarias para solucionar el problema | | | |
| 17. Los atributos creados en cada clase son los requeridos y están definidos adecuadamente | | | |
| 18. Los métodos creados en cada clase son los requeridos y están definidos adecuadamente | | | |
| 19. Utiliza herencia de forma lógica y funcional | | | |
| 20. Utiliza polimorfismo de forma lógica y funcional | | | |
| 21. El diagrama de clases creado resuelve el problema del proyecto | | | |
| 22. El diagrama lógico de la base de datos es acorde al diagrama de clases | | | |
| 23. El diccionario de datos de la base de datos defina las tablas correctamente | | | |
| Total: | | | |
| Simbología: CC: correcto y completo CI: correcto pero incompleto II: incorrecto e incompleto | Puntos Totales: | 46 | |
| | Puntos obtenidos: | | |
| | Porcentaje obtenido: | | |
| | Calificación: | | |

Observaciones:

PROYECTO FASE 2. Valor 20%

Escala numérica de evaluación

Objetivo de la Fase 2: Codificar aplicaciones de escritorio orientadas a objetos y sus elementos según los requerimientos establecidos por la clientela.

Persona participante:

| Lista de rasgos observables | Criterios | | |
|---|-----------|----|----|
| | CC | CI | II |
| | 2 | 1 | 0 |
| En cuanto a la presentación | | | |
| 1. Incluye portada con los datos requeridos | | | |
| 2. Incluye documentación externa de clases la cual debe incluir una descripción de atributos y métodos de cada clase desarrollada | | | |
| 3. Hace las correcciones de la fase 1 | | | |
| En cuanto al desarrollo. FASE 2 | | | |
| 4. Define las 4 capas de la aplicación | | | |
| 5. Respeta la funcionalidad definida en clases para cada capa | | | |

| | | | |
|--|-----------------------------|----|--|
| 6. Diseña los formularios siguiendo los estándares de definición de nombres establecido | | | |
| 7. Crea los formularios con los controles y propiedades correctas. | | | |
| 8. Define variables correctamente | | | |
| 9. Utilizas estructuras de control según lo requerido | | | |
| 10. Realiza manejo de excepciones dentro de la solución | | | |
| 11. Crea la base de datos de la aplicación | | | |
| 12. Utiliza las clases de ADO.Net adecuadamente | | | |
| 13. Realiza la conexión correctamente con la base de datos. | | | |
| 14. Inserta correctamente los datos solicitados. | | | |
| 15. Modifica los datos correctamente según lo solicitado. | | | |
| 16. Elimina los datos correctamente según lo solicitado. | | | |
| 17. Crea las funciones y métodos según lo requerido. | | | |
| 18. Crea las clases correctamente. | | | |
| 19. Las clases contiene los atributos necesarios | | | |
| 20. Los atributos definidos para las clases son del tipo adecuado. | | | |
| 21. Crea las propiedades adecuada a cada clase. | | | |
| 22. Crea los métodos requeridos según lo requerido. | | | |
| 23. Utiliza herencia correctamente | | | |
| 24. Utiliza polimorfismo correctamente. | | | |
| 25. Existe modularidad en la aplicación creada. | | | |
| 26. La visibilidad de los objetos es apropiada para la solución de proyecto. | | | |
| 27. Realiza sobrecarga de métodos. | | | |
| 28. Utiliza constructores dentro del proyecto. | | | |
| 29. Realiza las instancias de los objetos correctamente. | | | |
| 30. Utiliza buenas prácticas de programación. | | | |
| 31. Utiliza prefijos y sufijos par la declaración de los objetos y variables. | | | |
| 32. La aplicación es amigable | | | |
| 33. Es fácil de usar, aporta en cuanto a experiencia de usuario. | | | |
| 34. Contiene todos los elementos necesarios para resolver el proyecto eficientemente. | | | |
| 35. Resuelve el problema solicitado. | | | |
| 36. La solución presentada es funcional y acorde a lo solicitado. | | | |
| 37. Realiza la documentación interna de los métodos y clases | | | |
| Total: | | | |
| Simbología: CC: correcto y completo CI: correcto pero incompleto II: incorrecto e incompleto | Puntos Totales: | 74 | |
| | Puntos obtenidos: | | |
| | Porcentaje obtenido: | | |
| | Calificación: | | |

Observaciones:

PROYECTO FASE 3. Valor 20%

Escala numérica de evaluación

Objetivo de la Fase 3: Codificar aplicaciones Web orientadas a objetos y sus elementos según los requerimientos establecidos por la clientela.

Persona participante:

| Lista de rasgos observables | | Criterios | | |
|--|-----------------------------|-----------|----|----|
| | | CC | CI | II |
| | | 2 | 1 | 0 |
| En cuanto al desarrollo. FASE 3 | | | | |
| 1. Utiliza las 4 capas de la aplicación | | | | |
| 2. Respeta la funcionalidad definida en clases para cada capa | | | | |
| 3. Diseña los formularios web siguiendo los estándares de definición de nombres establecido | | | | |
| 4. Crea los formularios web con los controles y propiedades correctas. | | | | |
| 5. Realiza manejo de excepciones dentro de la solución | | | | |
| 6. Inserta correctamente los datos solicitados. | | | | |
| 7. Modifica los datos correctamente según lo solicitado. | | | | |
| 8. Elimina los datos correctamente según lo solicitado. | | | | |
| 9. Realiza las instancias de los objetos correctamente. | | | | |
| 10. El diseño de la página web es acorde a lo solicitado. | | | | |
| 11. Diseña una master Page para la aplicación web | | | | |
| 12. Utiliza variables de sesión o viewstate en los casos requeridos | | | | |
| 13. Utiliza el fichero web.config para la configuración de la aplicación. | | | | |
| 14. Utiliza buenas prácticas de programación. | | | | |
| 15. Utiliza controles de validación correctamente | | | | |
| 16. La aplicación es amigable | | | | |
| 17. Es fácil de usar. | | | | |
| 18. Contiene todos los elementos necesarios para resolver el proyecto eficientemente. | | | | |
| 19. Resuelve el problema solicitado. | | | | |
| 20. La solución presentada es funcional y acorde a lo solicitado. | | | | |
| 21. Realiza la documentación interna de los métodos y clases | | | | |
| Total: | | | | |
| Simbología: CC: correcto y completo CI: correcto pero incompleto II: incorrecto e incompleto | Puntos Totales: | 42 | | |
| | Puntos obtenidos: | | | |
| | Porcentaje obtenido: | | | |
| | Calificación: | | | |

Observaciones: