

CARRERA INGENIERÍA INFORMÁTICA
ASIGNATURA PRÁCTICA PROFESIONAL DE 2DO AÑO
CURSO 2024

GUÍA:

Tabla de Contenido

1. Introducción.....	1
1.1. Objetivos Generales.....	1
1.2. Conocimientos esenciales a adquirir:	2
1.3. Habilidades básicas a dominar:	2
2. Contenidos a desarrollar	2
2.1. Planteamiento general del problema a resolver	2
2.2. Hoja de trabajo.....	3
3. Informe	4
3.1. Estructura del informe	4
4. Uso del GIT	5
5. Defensa y Evaluación del Proyecto	5
Anexos	6

1. Introducción

La asignatura *Práctica Profesional de Segundo Año* forma parte del currículo base de la disciplina Práctica Profesional. Esta asignatura se desarrolla fundamentalmente a través del trabajo individual/por equipos de los estudiantes mediante el desarrollo de un Trabajo de Curso que constituye su forma de evaluación final.

La *Práctica Profesional de Segundo Año* se considera una asignatura integradora pues los estudiantes deben aplicar los conocimientos y desarrollar las habilidades de asignaturas del año actual, así como del año precedente.

1.1. Objetivos Generales

Los objetivos de la Práctica Profesional de 2do Año de Ing. Informática en el Plan E son:

- Aplicar de manera independiente y con profesionalidad lo aprendido a fin de proponer soluciones informáticas a una diversidad de problemas de acuerdo a los objetivos alcanzados hasta el año. Para cumplir este objetivo se puede hacer una aplicación de escritorio en java que automatice algún proceso de la empresa.
- Utilizar las tecnologías adecuadas para el desarrollo de soluciones informáticas.
- Consultar literatura en idioma extranjero para identificar los conceptos fundamentales y los trabajos relacionados con la solución informática que se proyecte desarrollar.
- Documentar la ejecución de proyectos de sistemas informáticos con artefactos ingenieriles adecuados en función de la complejidad y la naturaleza del proyecto.
- Analizar una situación con enfoque sistémico para evaluar la necesidad de soluciones informáticas que le generen un valor tangible o intangible.
- Desarrollar los procesos de gestión de software y sus infraestructuras tecnológicas atendiendo a las particularidades del proyecto profesional y asegurando su calidad.
- Desarrollar hábitos de disciplina, trabajo en equipo, independencia, creatividad, organización personal y responsabilidad ante el trabajo.

- Consolidar hábitos de auto-superación que posibiliten mantener un nivel profesional de acuerdo al acelerado desarrollo científico-técnico en la informática

1.2. Conocimientos esenciales a adquirir:

Los conocimientos propios de las demás disciplinas de la carrera sistematizados en el proceso de solucionar un problema de naturaleza informática. Métodos para la revisión de bibliografía científico-técnica y las herramientas automatizadas para su gestión como referencias bibliográficas. Elementos de Metodología de Investigación. Justificación y planteamiento de un problema con énfasis en los problemas de naturaleza técnica. Diseño metodológico de una práctica profesional. Normas generales para la redacción de informes científico-técnicos, presentación de resultados y su comunicación oral.

En particular:

- Análisis, diseño e implementación de una aplicación en un lenguaje de alto nivel (JAVA) conteniendo una interfaz gráfica capaz de responder a algunas responsabilidades del sistema definidas en el proyecto asignado.
- Diseño, uso e implementación de estructuras de datos propias y/o incluidas en el lenguaje de programación JAVA que permitan manipular los datos necesarios para el desarrollo de responsabilidades propias del trabajo de curso.
- Documentación del diseño de la solución obtenida

1.3. Habilidades básicas a dominar:

1. Integrar los conocimientos de la carrera en la concepción y desarrollo de una solución informática
2. Documentar la solución informática propuesta utilizando los artefactos ingenieriles de la especialidad
3. Identificar los conceptos fundamentales en el ámbito de un problema y una solución potencial
4. Justificar la existencia de un problema argumentando la existencia de una necesidad concreta
5. Plantear un objetivo general y sus objetivos específicos en función de metas medibles
6. Estructurar adecuadamente un informe científico-técnico
7. Realizar un análisis crítico en múltiples fuentes bibliográficas que recojan soluciones informáticas similares dado una problemática determinada
8. Redactar documentos científico-técnicos cumpliendo las normas de ortografía y redacción de la lengua
9. Utilizar un gestor de referencias para el almacenamiento y recuperación de recursos bibliográficos y documentales personales
10. Comunicarse adecuadamente de manera oral y escrita en múltiples auditorios
11. Escribir manuales que permitan instalar y operar las soluciones informáticas desarrolladas

En particular:

- Diseñar estructuras de datos y algoritmos complejos.
- Desarrollar y poner a punto de programas en lenguajes de alto nivel.
- Documentar programas

2. Contenidos a desarrollar

El Trabajo de Curso a desarrollar involucra directamente contenidos de asignaturas del año actual como Estructuras de Datos y Base de Datos, así como de asignaturas recibidas en años precedentes como Introducción a la Programación, Diseño y Programación Orientado a Objetos.

2.1. Planteamiento general del problema a resolver

Cada estudiante debe programar una aplicación (*front-end*) con interfaz gráfica que permita dar solución al problema asignado. Los estudiantes visitarán la entidad laboral de base o unidad docente

para recopilar información sobre el problema en cuestión que se le plantea para resolver. Se debe entregar como primera tarea una descripción del problema a resolver. Es deseable que la propuesta que se presente pertenezca a alguno de los sectores estratégicos siguientes:

- Turismo
- Industria biotecnológica y farmacéutica
- Electroenergético
- Producción de alimentos
- Construcciones
- Telecomunicaciones e informática
- Logística
- Redes hidráulicas y sanitarias
- Agroindustria azucarera
- Industria ligera
- Servicios técnicos profesionales

Las responsabilidades seleccionadas deben ser tratadas en la aplicación de forma tal que:

1. Se implemente la captación, modificación, eliminación y almacenamiento de los datos.
2. Se deben utilizar estructuras de datos creadas por el estudiante o del propio lenguaje de programación JAVA.
3. Se deben utilizar ficheros para el almacenamiento de los datos.
4. Debe considerarse adecuadamente el tratamiento de errores y excepciones.
5. Toda la codificación se debe realizar con control de versiones dejando el proyecto en el servidor de la universidad, donde se pueda revisar la participación individual de los integrantes del equipo en el desarrollo del proyecto.

2.2. Hoja de trabajo

A partir de lo anterior, se describen las tareas que deben ser ejecutadas como parte del proyecto

1. Requisitos
 - a. Definir los requisitos funcionales del sistema
 - b. Definir los casos de uso del sistema. Se debe obtener el Diagrama de casos de Uso
2. Diseño de la aplicación:
 - a. Identificar las clases necesarias, sus responsabilidades y la colaboración entre ellas. Agruparlas según patrón de diseño basado en tres capas: Interfaz, Negocio, Datos
 - b. Definir las relaciones entre clases. Se debe obtener el Diagrama de Clases
 - c. Describir las responsabilidades del sistema. Se debe definir el Diagrama de Actividades ó pseudocódigo de cada una de las funciones que intervienen en la responsabilidad seleccionada.
3. Programación de la aplicación
 - a. Implementar la interfaz gráfica que permita la captación de datos y la ejecución de las acciones necesarias
 - b. Diseñar e implementar las estructuras de datos adecuadas para el procesamiento de la información en cada caso
 - c. Implementar las responsabilidades del problema
4. Puesta a punto y prueba de la aplicación
 - a. Diseñar y aplicar casos de prueba que permitan chequear el funcionamiento de todas las opciones implementadas
5. Documentación del proceso
 - a. Todos los pasos desarrollados deben quedar adecuadamente documentados en un informe final que debe ser elaborado de acuerdo a los aspectos descritos en la sección

3. Informe

Como parte del trabajo de curso debe entregarse un informe escrito ajustado a la estructura que se describe a continuación. Para su evaluación se tendrá en cuenta la calidad y claridad con que se presente la información, poder de síntesis, ortografía y redacción. Es imprescindible su presentación para poder defender el proyecto.

El documento debe ser escrito en tercera persona del singular, en Arial 11 e interlineado simple, texto justificado.

3.1. Estructura del informe

El informe debe contener las secciones

- Presentación
- Resumen (Breve explicación del objetivo, la solución propuesta y los resultados obtenidos)
- Índice
- Introducción (Descripción de la situación problemática y objetivos del trabajo de curso)
- Descripción de la Solución Propuesta
 - Requisitos
 - Diseño
 - Programación
 - Pruebas
- Conclusiones (Respondiendo a los objetivos planteados, nivel de cumplimiento de estos, elementos importantes a resaltar que caractericen los resultados)
- Recomendaciones
- Referencias Bibliográficas
- Anexos (Modelo lógico y físico de los datos, diagrama entidad-relación, otros diagramas, tablas, etc. que sean útiles para explicar o lograr un mejor entendimiento de las ideas transmitidas en el cuerpo del informe. Todo lo que se presente aquí debe ser referenciado en el texto)

La sección “Descripción de la Solución Propuesta” debe incluir

- Requisitos:
 - Resultados obtenidos en el punto 1.a de la Hoja de Trabajo (sección 2.4)
 - Diagrama de Casos de Usos
- Diseño:
 - Resultados obtenidos en el punto 1.c de la Hoja de Trabajo (sección 2.4)
 - Diagrama de clases con su explicación y consideraciones hechas para su obtención
 - Descripción de una de las responsabilidades utilizando los diagramas de actividad o pseudocódigo necesarios.
- Programación
 - Consideraciones hechas para la implementación de la interfaz gráfica y de las diferentes responsabilidades
 - Estructuras de datos utilizadas para el tratamiento de los datos en memoria interna y fundamentación de las mismas
 - Manual de usuario donde se describan las funciones que realiza la aplicación y como desarrollarlas a través del sistema. Incluir imágenes de cuadros de diálogos, etc.
- Pruebas
 - Especificación de los casos de prueba diseñados y resultados obtenidos con cada uno

La sección de “Referencias bibliográficas” debe contener la referencia a la bibliografía consultada en relación tanto a temas de la especialidad como a otros necesarios para el desarrollo del proyecto.

4. Uso del GIT

Para realizar los controles intermedios del proyecto, los responsables del control de la práctica utilizarán las versiones del proyecto que se encuentren en la plataforma de control de versiones GIT de la UCLV (<https://git.uclv.edu.cu>). Para ello, los estudiantes deben crear un proyecto en el repositorio GIT y mantener actualizadas las versiones del mismo utilizando un cliente como TortoiseGIT. Es imprescindible que el estudiante garantice mantener en el proyecto GIT la siguiente información:

- Descripción del objeto de automatización (orientación del proyecto de base de datos especificando las responsabilidades a desarrollar específicamente en la práctica profesional)
- Descripción del proyecto a realizar, identificando los aspectos a desarrollar y el nivel de cumplimiento actual
- Proyecto JAVA del sistema en desarrollo

Se entregarán materiales y se capacitará mediante actividades presenciales, el trabajo con el control de versiones en GIT.

5. Defensa y Evaluación del Proyecto

- La evaluación final del proyecto se desarrollará en la última semana del período concentrado de prácticas laborales, en la fecha establecida por los responsables de la práctica. La defensa se realizará ante un tribunal.
- Es importante haber entregado el informe del proyecto con días de antelación a la fecha de la defensa, a través de una tarea evaluativa en la plataforma Moodle.
- Sobre la discusión final
 - Los estudiantes deben llevar los proyectos completos incluyendo el código fuente y el juego de datos para la ejecución de la aplicación informática.
 - Los estudiantes deben desarrollar una exposición de 15 minutos donde se describa el problema a resolver, los criterios fundamentales tenidos en cuenta para el diseño y la programación de la aplicación, así como mostrar el funcionamiento de esta,
 - El tribunal formulará preguntas que permitan demostrar al estudiante, un dominio adecuado de lo realizado, así como habilidades para realizar cualquier modificación al código,
 - Durante la exposición, los estudiantes deben mostrar habilidades comunicativas y estar adecuadamente vestidos.
 - La práctica será desarrollada por equipos, a lo sumo, por 3 estudiantes. Debe quedar claramente definido qué rol y tareas asociadas realizó cada cual.
- Criterios para la evaluación final. El tribunal otorgará la calificación final a partir de la calidad mostrada en los diferentes aspectos que se muestran en la lista de chequeo del **Anexo 1**. Los criterios para cada aspecto se pueden formar a partir de lo expuesto en el informe, el código fuente de la aplicación, la exposición del estudiante y/o las preguntas realizadas por el tribunal. Es importante aclarar que la nota no es colectiva para todos los integrantes del equipo, sino debe ajustarse a los aspectos mostrados por cada estudiante en específico.

Anexos

Anexo 1. Aspectos a evaluar en la Práctica Profesional de Segundo Año

Aspectos a evaluar	Donde se evalúa (I, C, E, P)*	Influencia (%)	Evaluación (E, MB, B, R, M)
Informe		15	
• Adecuación a la estructura y contenido establecidos	I	40	
• Calidad general del informe	I	40	
• Ortografía y redacción	I	20	
Requisitos		20	
• Definición correcta de requisitos funcionales	I / E / P	50	
• Diagrama de Casos de Uso. Determinación de responsabilidades y actores	I / E / P	50	
Diseño		20	
• Definición correcta de requisitos funcionales	I / E / P	20	
• Diagrama de Casos de Uso. Determinación de responsabilidades y actores responsables.	I / E / P	20	
• Definición de clases según patrón de diseño basado en 3 capas. (valorar además la correspondencia con el Diagrama de Casos de Usos, ej. los actores definen clases pertenecientes a la capa de negocio, los casos de uso constituyen responsabilidades en las clases de la capa de negocio, etc.)	I / E / P	20	
• Diagrama de clases. Valorar correspondencia con la colaboración entre clases especificada	I / E / P	20	
• Descripción de una responsabilidad principal del sistema utilizando un conjunto de diagramas de actividad o el pseudocódigo de cada una de las funciones necesarias.	I / E / P	20	
Programación		35	
• Uso adecuado de las estructuras de datos	C / E / P	20	
• Tratamiento correcto de las formas de organización de la información en memoria externa	C / E / P	20	
• Funcionamiento correcto de las opciones del sistema	C / E / P	20	
• Tratamiento adecuado de errores y excepciones	C / E / P	20	
• Habilidades del estudiante para modificar el código de acuerdo a nuevas necesidades	P / C	20	
Prueba		15	
• Definición y uso adecuado de casos de pruebas	I	100	
Exposición		15	
• Dominio del contenido	E	70	
• Forma de expresión, fluidez, vocabulario	E	30	
Evaluación Final			

* (I): Informe, (C): Código, (E): Exposición, (P): Preguntas