



Facultad: INGENIERÍA

Programa: INGENIERÍA ELECTRÓNICA

1. Identificación del curso

Nombre: Programación II

Área: Básica

Código: BEINGELO92

Número de créditos: 3

Horas de
acompañamiento
directo:

64

Horas de trabajo
independiente:

80

Total Horas:

144

Carácter del curso (Teórico, práctico o teórico práctico): Teórico práctico

Componente Básico o complementario: Básico

Requisito: Programación I

Unidad responsable del microdiseño: Ingeniería electrónica

2. Presentación del curso

El ejercicio de la ingeniería en la sociedad de la información y del conocimiento, requiere de habilidades y competencias como: resolución de problemas complejos, automatización de tareas, análisis de datos, desarrollo de aplicaciones, integración de sistemas informáticos y hardware, innovación tecnológica, comunicación, entre otras. En este sentido, el curso de programación II ofrece un espacio académico, que permite el fortalecimiento de dichas competencias, privilegiando la innovación y el desarrollo de aplicaciones que involucran sistemas embebidos y distribuidos.

El presente curso está dividido en tres grandes partes. Inicialmente, se afianzarán habilidades de programación y especificación de sistemas informáticos monolíticos, empleando nociones de ingeniería de software. Luego, se abordará el diseño y desarrollo de aplicaciones distribuidas, lo cual permitirá aplicar los saberes construidos hasta el momento a entornos reales de operación. Finalmente, el curso se enfocará en el desarrollo de servicios telemáticos que integran informática, tecnologías de la comunicación y sistemas embebidos.



3. Justificación

El programa de ingeniería electrónica reconoce la necesidad de integrar fundamentos en informática y programación en el proceso formativo de sus estudiantes, como respuesta a la creciente demanda de soluciones tecnológicas innovadoras del sector. En este contexto, la competencia informática y digital, deja de ser de una simple capacidad técnica; para convertirse en una competencia esencial para el éxito profesional. En la actualidad, el mercado laboral exige que los ingenieros electrónicos posean habilidades analíticas y de comprensión intuitiva de los sistemas digitales para proponer soluciones a problemas complejos, que en su mayoría involucran: datos, dispositivos electrónicos, y grupos de trabajo multidisciplinarios. Entendiendo estas dinámicas, este curso se diseña con el objetivo de formar ingenieros de vanguardia, aptos para afrontar los desafíos del diseño, desarrollo, implementación y mantenimiento de sistemas telemáticos avanzados, lo cual permitirá marcar una diferencia significativa en su perfil profesional, mientras se contribuye al desarrollo de la región y del país a través de un nuevo paradigma tecnológico.

4. Competencias

1. Capacidad de identificar, formular y resolver problemas complejos de ingeniería mediante la aplicación de principios de ingeniería, ciencias y matemáticas.
2. Capacidad para comunicarse de manera efectiva con una variedad de audiencias.
3. Capacidad de reconocer responsabilidades éticas y profesionales en situaciones de ingeniería y emitir juicios informados, que deben considerar el impacto de las soluciones de ingeniería en contextos globales, económicos, ambientales y sociales.
4. Capacidad para funcionar de manera efectiva en un equipo cuyos miembros juntos brindan liderazgo, crean un entorno colaborativo e inclusivo, establecen metas, planifican tareas y cumplen objetivos.
5. Capacidad de desarrollar y realizar experimentos apropiados, analizar e interpretar datos y utilizar el juicio de ingeniería para sacar conclusiones.
6. Capacidad de adquirir y aplicar nuevos conocimientos según sea necesario, utilizando estrategias de aprendizaje adecuadas.

5. Resultados de aprendizaje, actividades académicas y estrategias de evaluación

Resultados de Aprendizaje	Actividades Académicas	Estrategias de Evaluación
Justifica el diseño y especificación de aplicaciones informática empleando técnicas actuales de ingeniería del software.	<ul style="list-style-type: none">• Clases magistrales.• Talleres individuales y colectivos de diseño.• Laboratorios y prácticas.	<ul style="list-style-type: none">• Evaluación escrita.• Actividades en clase.• Presentaciones y sustentaciones orales.• Evaluaciones cortas.
Desarrolla aplicaciones monolíticas que analizan	<ul style="list-style-type: none">• Clases magistrales.	<ul style="list-style-type: none">• Evaluación escrita.

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



MICRODISEÑO CURRICULAR

CÓDIGO

MI-FOR-FO-34

VERSIÓN

2

VIGENCIA

2022

Página

3 de 10

información empleando librerías y paquetes básicos de python, emplea datos ingresados por el usuario y datos persistentes almacenados localmente.	<ul style="list-style-type: none"> • Aprendizaje basado en problemas (ABP). • Laboratorios y prácticas. 	<ul style="list-style-type: none"> • Actividades en clase. • Presentaciones y sustentaciones orales. • Evaluaciones cortas.
Desarrolla aplicaciones distribuidas que recolectan y analiza información de múltiples fuentes empleando librerías y paquetes avanzados de python, así como base de datos relacionales y no relacionales.	<ul style="list-style-type: none"> • Clases magistrales. • Aprendizaje basado en problemas (ABP). • Laboratorios y prácticas. 	<ul style="list-style-type: none"> • Evaluación escrita. • Actividades en clase. • Presentaciones y sustentaciones orales. • Evaluaciones cortas.
Desarrolla servicios telemáticos avanzados que integran componentes distribuidos, base de datos, y clientes heterogéneos.	<ul style="list-style-type: none"> • Clases magistrales. • Aprendizaje mediante proyectos (AMP). • Laboratorios y prácticas. 	<ul style="list-style-type: none"> • Evaluación escrita. • Actividades en clase. • Presentaciones y sustentaciones orales. • Evaluaciones cortas.
Desarrollar programas en Python para realizar experimentos computacionales, analizar los resultados obtenidos, e interpretar los datos con juicio ingenieril, aplicando estrategias de aprendizaje autónomo para mejorar continuamente los programas implementados.	<ul style="list-style-type: none"> • Aprendizaje mediante proyectos. • Trabajo en grupo. 	<ul style="list-style-type: none"> • Presentaciones y sustentaciones orales. • Laboratorios y prácticas.

6. Evaluación general del curso

Resultado de aprendizaje	Desempeño deseado				
Justifica el diseño y especificación de aplicaciones informática empleando técnicas actuales de ingeniería del software.	El estudiante es capaz justificar a través de argumentos concisos y verificables el diseño de una aplicación informática empleando lenguaje unificado de modelado (UML), orientación a objetos, y metodologías propias de ingeniería del software.				
	Completa mente alcanzado (100%)	Alcanzado en alto grado (70-90%)	Alcanzado de manera aceptable (50-70%)	Aun no alcanzado (10-50%)	Aun no intentado (0-10%)
Desarrolla aplicaciones monolíticas que analizan información suministrada por el usuario y de repositorios locales,	El estudiante es capaz de desarrollar aplicaciones monolíticas a partir de un requerimiento dado. Realiza su diseño y especificación, a través de UML y paradigmas propios de ingeniería del software. También, posee la habilidad de crear representaciones estructurales y de comportamiento básicas, como: diagrama de clases, paquetes,				

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



MICRODISEÑO CURRICULAR

CÓDIGO

MI-FOR-FO-34

VERSIÓN

2

VIGENCIA

2022

Página

4 de 10

empleando librerías y paquetes básicos de python.	objetos, componentes, actividades, secuencia, comunicación, interacción, casos de uso, estado, entre otros.				
	Completa- mente alcanzado (100%)	Alcanzado en alto grado (70-90%)	Alcanzado de manera aceptable (50-70%)	Aun no alcanzado (10-50%)	Aun no intentado (0-10%)
Desarrolla aplicaciones distribuidas que recolectan y analiza información de múltiples fuentes empleando librerías y paquetes avanzados de python, así como base de datos relacionales y no relacionales.	El estudiante es capaz de desarrollar aplicaciones distribuidas a partir de un requerimiento dado. Realiza su diseño y especificación, a través de UML y paradigmas propios de ingeniería del software. También, posee la habilidad de crear representaciones estructurales y de comportamiento avanzadas, como: diagrama de clases, paquetes, objetos, componentes, actividades, secuencia, comunicación, interacción, casos de uso, estado, entre otros.				
	Completa- mente alcanzado (100%)	Alcanzado en alto grado (70-90%)	Alcanzado de manera aceptable (50-70%)	Aun no alcanzado (10-50%)	Aun no intentado (0-10%)
Desarrolla servicios telemáticos avanzados que integran componentes distribuidos, base de datos, y clientes heterogéneos.	El estudiante diseña, formula y ejecuta proyectos que involucran el desarrollo de servicios telemáticos, a partir de la integración de componentes distribuidos, base de datos, clientes heterogéneos, y sistemas embebidos.				
	Completa- mente alcanzado (100%)	Alcanzado en alto grado (70-90%)	Alcanzado de manera aceptable (50-70%)	Aun no alcanzado (10-50%)	Aun no intentado (0-10%)
Desarrollar programas en Python para realizar experimentos computacionales, analizar los resultados obtenidos, e interpretar los datos con juicio ingenieril, aplicando estrategias de aprendizaje autónomo para mejorar continuamente los programas implementados.	El estudiante desarrolla experimentos completos en Python, analizando de manera adecuada los resultados y utilizando criterios de ingeniería para sacar conclusiones precisas. Además, demuestra una mejora continua en sus habilidades mediante la implementación de nuevas técnicas y estrategias de aprendizaje autónomo, integrando bibliotecas avanzadas o algoritmos optimizados.				
	Completa- mente alcanzado (100%)	Alcanzado en alto grado (70-90%)	Alcanzado de manera aceptable (50-70%)	Aun no alcanzado (10-50%)	Aun no intentado (0-10%)

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



SC 7184-1



SA-CERE 987526



OS-CER 987555



7. Unidades temáticas, estrategias didácticas y tiempo asignado

No.	Unidades y contenidos	Estrategias didácticas	Horas				
			Acompañamiento directo			Trabajo Independiente	Total
			Teóricas	Teórico-Prácticas	Prácticas	Independiente	
1	Introducción al desarrollo de aplicaciones monolíticas.	Clases magistrales, ejemplos y talleres.	2		2	5	9
2	Librerías básicas.	Clases magistrales, ejemplos y talleres.	2		2	5	9
3	Manejo de archivos	Clases magistrales, ejemplos y talleres.	2		2	5	9
4	Proyecto de corte I	Exposición y sustentación.	2		2	5	9
5	Introducción al lenguaje de modelado (UML).	Clases magistrales, ejemplos y talleres.	2		2	5	9
6	Introducción al desarrollo de sistemas distribuidos.	Clases magistrales, ejemplos y talleres.	2		2	5	9

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



SC 7184-1



SA-CERE 987526



OS-CER 987555



MICRODISEÑO CURRICULAR

CÓDIGO

MI-FOR-FO-34

VERSIÓN

2

VIGENCIA

2022






Página

6 de 10

7	Base de datos relacionales y no relacionales.	Clases magistrales, ejemplos talleres. y	2		2	5	9
8	Creación de contenidos con HTML	Clases magistrales, ejemplos talleres. y	2		2	5	9
9	Creación de contenidos con CSS3 y javascript	Clases magistrales, ejemplos talleres. y	2		2	5	9
10	Programación web con Django	Clases magistrales, ejemplos talleres. y	2		2	5	9
11	Proyecto de corte II	Exposición y sustentación.	2		2	5	9
12	Introducción al desarrollo de servicios telemáticos.	Clases magistrales, ejemplos talleres. y	2		2	5	9
13	Introducción a GIT	Clases magistrales, ejemplos talleres. y	2		2	5	9
14	Servicios telemáticos (Django y MongoDB)	Clases magistrales, ejemplos talleres. y	2		2	5	9
15	Servicios telemáticos avanzados	Clases magistrales,	2		2	5	9

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.

	UNIVERSIDAD SURCOLOMBIANA FORMACIÓN					   	
	MICRODISEÑO CURRICULAR					<small>SC 7184-1 SA-CERES 987526 OS-CER 987555</small>	
CÓDIGO	MI-FOR-FO-34	VERSIÓN	2	VIGENCIA	2022	Página	7 de 10

		ejemplos talleres.	y				
16	Proyecto final	Exposición sustentación.	y	2		2	5
Totales				32		32	80
Total				64		80	144

*Entiéndase por práctica las actividades académicas realizadas en espacios formativos, donde se contrastan los fundamentos teóricos y prácticos. **Especificar la naturaleza de la práctica (Clínica, Pedagógica, Laboratorio, etc.)

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



8. Referencias bibliográficas

- Oliver, R. (2023). Python QuickStart Guide: The Simplified Beginner's Guide to Python Programming Using Hands-On Projects and Real-World Applications. Clydebank Media LLC.
- Matthes, E. (2023). Python Crash Course, 3rd Edition. No Starch Press.
- Sweigart, A. (2019). Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners. No Starch Press.
- Ramalho, L. (2021). Fluent Python: Clear, Concise, and Effective Programming.
- Lutz, M. (2013). Learning Python (5th ed.). O'Reilly Media.
- Robbins, P. (2023). Python Programming for Beginners: The Complete Guide to Mastering Python in 7 Days with Hands-on Exercises - Top Secret Coding Tips to Get an Unfair Advantage and Land Your Dream Job!
- McKinney, W. (2022). Python for data analysis: Data Wrangling with Pandas, NumPy, and Jupyter. O'Reilly Media.
- Barry, P. (2023). Head first Python: A Brain-Friendly Guide. O'Reilly Media.
- Fowler, M. (2004). UML distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley Professional.
- Miles, R., & Hamilton, K. (2008). Learning UML 2.0. O'Reilly Media.
- Pilone, D., & Pitman, N. (2005). UML 2.0 in a Nutshell: A Desktop Quick Reference (1st ed.). O'Reilly Media.
- Samek, M. (2008). Practical UML State charts in C/C++, Second Edition: Event-Driven Programming for Embedded Systems.
- Rumbaugh, J., Jacobson, I., & Booch, G. (1998). The unified modeling language reference manual.
- Schmuller, J. (2004). Sams Teach Yourself UML in 24 Hours, Complete Starter Kit (3rd Edition).
- Dennis, A. R., Wixom, B. H., & Tegarden, D. P. (2016). Systems Analysis and Design: An Object-Oriented Approach with UML (5th edition).
- Dennis, A. R., Wixom, B. H., & Roth, R. M. (2016). Systems Analysis and Design (6th edition).
- Martin, R. C. (2018). Clean architecture: A Craftsman's Guide to Software Structure and Design. Pearson Professional.
- Raúl Gonzáles Duque. "Python para todos". <http://mundogeek.net/tutorial-python/> . 2020
- Jon Vadillo Romero. "Aprende Python de cero a experto". <https://dev.to/jvadillo/aprende-python-desde-cero-a-experto-4eej>. Editorial LeanPub. 2020
- https://www.w3schools.com/python/python_arrays.asp



SC 7384-1

SA-CER 167728

OS-CER 597000

9. Trazabilidad de la evaluación del microdiseño

Fecha de evaluación actualización y aprobación por el comité de currículo (número de acta)	Modificación	Justificación	Responsables
	Se realiza una actualización al último formato MI-FOR-FO-34, versión 2, año 2022	Es necesario actualizar para atender la solicitud a el registro calificado	Albeiro Cortés Cabezas
30/01/2024	Se colocó el código de la asignatura. Se actualizaron las competencias según ABET. Se actualizan actividades académicas de los resultados de aprendizaje.	Revisión y actualización	Martin Bravo Obando y Fernando Barrera.

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



Guía para diligenciar el Microdiseño Curricular

1. Diligenciar la información solicitada en el encabezado, relacionada con la Facultad, el Programa, nombre del curso y el área a la que pertenece el curso (si existen áreas).
2. Incluir el código del curso, los cuales inician con B cuando son cursos del componente básico, o F cuando son del componente complementario flexible, seguidos de la letra E cuando son específicas del programa o F cuando son de Facultad o I cuando son Institucionales.
3. Registrar el número de créditos académicos del curso o asignatura, basado en el plan de estudios aprobado por el Ministerio de Educación Nacional (MEN).
4. Registrar el número de horas de acompañamiento directo y número de horas de trabajo independiente del estudiante. El nuevo SACES solicita dicha información en horas teóricas, teórico-prácticas y prácticas.
5. Registrar el número de total de horas, el cual es la sumatoria de los puntos 4, tenga en cuenta que el número total de horas del curso debe ser coherente con el número de créditos. Recuerde que 1 crédito equivale a 48 horas para un periodo académico y el número de horas presenciales, dependerá de la naturaleza del curso y del número de semanas del periodo lectivo.
6. Escribir el nombre de la unidad académica responsable de la actualización y evaluación del currículo, así como si el curso pertenece al componente básico o complementario flexible y el requisito para cursar la asignatura.
7. Elaborar la presentación del curso, la justificación y sustentar la coherencia entre los resultados de aprendizaje del curso, del programa y del perfil de egreso, en consonancia con el componente teleológico de la Universidad.
8. Completar la información del punto 5 (tabla), la cual debe guardar coherencia entre: los resultados de aprendizaje, las estrategias de enseñanza y el sistema de evaluación.
9. Completar la información del punto 6, especificando las formas o instrumentos que se emplearán para evaluar los resultados de aprendizaje adquiridos por el estudiante.
10. Completar la información del punto 7 (tabla). No olvide que el total de horas debe ser coherente con el número de créditos del curso. En las estrategias didácticas, enuncie las formas como se evidenciará el trabajo independiente del estudiante, que serán concertadas en la primera semana de clases.
11. Escribir las referencias bibliográficas empleadas para el desarrollo del curso. Pueden ser recursos impresos, recursos audiovisuales o tecnológicos, así como materiales de laboratorio (equipos, sustancias químicas, etc.). Incluir por lo menos una referencia bibliográfica de las bases de datos con las que cuenta la Universidad Surcolombiana, una referencia bibliográfica en inglés y una referencia bibliográfica de las producciones académicas de intelectuales de Latinoamérica.
12. Diligenciar en el punto 9 (tabla) la trazabilidad de la evaluación del microdiseño, el cual incluye, fechas de evaluación, actualización y aprobación por el comité de currículo, modificación, justificación y responsables.