

**FACULTAD DE CIENCIAS E INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas e Informática**

SÍLABO

# DATOS GENERALES

|  |  |  |
| --- | --- | --- |
| **1.0** | Asignatura | Programación III |
| **1.1** | Programa de Estudios | Ingeniería de Sistemas e Informática |
| **1.2** | Área/ sub área | Formación Especializada |
| **1.3** | Código | ISI-FE5307 |
| **1.4** | Condición | Obligatorio |
| **1.5** | Créditos | 4 |
| **1.6** | Número de horas de teoría | 3 |
| **1.7** | Número de horas de práctica | 2 |
| **1.8** | Pre-requisitos | ISI-FE4304 |
| **1.9** | Ciclo | V |
| **1.10** | Semestre académico | 2025–I |
| **1.11** | Equipo Docente | Docente1  [docente1@uch.edu.pe](mailto:docente1@uch.edu.pe)  Docente2  [docente2@uch.edu.pe](mailto:docente2@uch.edu.pe) |

# SUMILLA

Esta asignatura corresponde al área de estudios especializados, su carácter es teórico práctico. Se propone que el estudiante aplique el enfoque de programación orientada a objetos con el lenguaje de programación Java desarrollando programas de diversa aplicación. Abarca los siguientes temas principales: Fundamentos de Java; Clases y Objetos; Encapsulamiento; Herencia y polimorfismo; Persistencia.

# COMPETENCIAS

**3.1. COMPETENCIA CLAVE**

Desarrolla soluciones de negocios innovadores que contribuya a la toma de decisiones en las organizaciones; identificando requerimientos específicos de la organización y utilizando metodologías, herramientas y lenguajes de programación en plataformas emergentes con actitud ética y proactiva.

**3.2. UNIDAD DE COMPETENCIA**

Desarrolla aplicaciones en diversas plataformas utilizando lenguajes y herramientas de desarrollo emergentes

# RESULTADOS DEL APRENDIZAJE DE LA ASIGNATURA

Al finalizar la asignatura, el estudiante será capaz de diseñar y crear clases y objetos aplicando principios de encapsulamiento, utilizando modificadores de acceso y constructores para inicializar objetos. Además, implementará herencia y polimorfismo para crear programas flexibles y reutilizables, aprovechando superclases, subclases, e interfaces. Por último, el estudiante gestionará el almacenamiento de datos en aplicaciones Java utilizando técnicas de persistencia, conectando a bases de datos mediante JDBC y manejando excepciones para asegurar la robustez del programa.

# CONTENIDOS

|  |  |  |
| --- | --- | --- |
| **UNIDAD DIDÁCTICA I: Fundamentos de Java; Clases y Objetos**  **Logro o resultado de aprendizaje de la unidad:**  Al finalizar esta unidad, el estudiante será capaz de comprender y aplicar los fundamentos del lenguaje de programación Java, incluyendo la sintaxis básica, la estructura de un programa Java, y el uso de un entorno de desarrollo integrado (IDE). Además, podrá diseñar y crear clases y objetos, comprendiendo la importancia de la orientación a objetos y utilizando correctamente variables, operadores y estructuras de control. | | |
| **Semana** | **Contenidos Teórico-Prácticos** | |
| 1 | T: Presentación del sílabo. Introducción a Java; características del Lenguaje de Programación Orientado a Objetos, estructura de un programa en Java. Tipos de datos.  P: Creación de programas en Java. | |
| 2 | T: Estructuras de Programación; declaración y uso de variables; operadores aritméticos, relacionales y lógicos. Estructuras de control condicionales y repetitivas; uso de bloques de código.  P: Desarrollo de programas con estructuras de control, creación de un menú interactivo usando Switch. | |
| 3 | T: Conceptos de programación orientada a objetos; definición de clases objetos, métodos y tipos de métodos. Instanciación de un objeto  P: Creación de programas de clases con atributos y métodos, instanciación de objetos | |
| 4 | T: Sobrecarga de métodos, constructores y destructores. Aplicaciones de uso  P: implementación y uso de programas sobrecargados, constructores con y sin parámetros. | |
| **Producto o evidencia de aprendizaje:**  Incluye programas, ejercicios con operadores y estructuras de control, elaboración de clases representando la estructura y relaciones de objetos desarrollados. | | |
| **Instrumentos de evaluación del logro de aprendizaje de la unidad:** | | * Rúbrica. * Resolución de casos prácticos. |

|  |  |  |
| --- | --- | --- |
| **UNIDAD DIDÁCTICA II: Encapsulamiento**  **Logro o resultado de aprendizaje de la unidad:**  Al finalizar esta unidad, el estudiante será capaz de aplicar principios de encapsulamiento en el diseño de clases en Java, utilizando modificadores de acceso para controlar la visibilidad de atributos y métodos. Será capaz de proteger los datos dentro de las clases, asegurando una interacción segura y controlada entre los componentes del programa. | | |
| **Semana** | **Contenidos Teórico-Prácticos** | |
| 5 | T: Principios de encapsulamiento; modificadores de acceso (public, private, protected); uso de this.  P: Implementación de clases con atributos privados; desarrollo de métodos para acceder y modificar atributos privados. | |
| 6 | T: Métodos estáticos y variables estáticas; diferencias entre miembros estáticos y no estáticos.  P: Ejercicios de implementación y uso de métodos y variables estáticas en clases; creación de un contador de instancias de una clase usando variables estáticas | |
| 7 | T: Manejo de excepciones en Java; tipos de excepciones (checked y unchecked); bloques try, catch, finally; palabra clave throws.  P: Implementación de manejo de excepciones; desarrollo de programas que capturen y manejen diferentes tipos de excepciones; creación de clases de excepciones personalizadas | |
| 8 | **EXAMEN PARCIAL** | |
| **Producto o evidencia de aprendizaje:**  Implementación de clases con atributos privados y métodos públicos para acceder y modificar dichos atributos, ejercicios utilizando encapsulamiento para proteger y controlar el acceso a los datos de los objetos. | | |
| **Instrumentos de evaluación del logro de aprendizaje de la unidad** | | * Rúbrica. * Resolución de casos prácticos. * Cuestionario |

|  |  |  |
| --- | --- | --- |
| **UNIDAD DIDÁCTICA III: Herencia y polimorfismo**  **Logro o resultado de aprendizaje de la unidad:**  Al finalizar esta unidad, el estudiante será capaz de implementar herencia y polimorfismo en sus programas Java, creando jerarquías de clases que permitan la reutilización y extensión del código. Será capaz de utilizar superclases y subclases, sobrescribir métodos, y aplicar el principio de polimorfismo para que objetos de diferentes clases respondan de manera adecuada a las mismas llamadas de método. | | |
| Semana | **Contenidos Teórico-Prácticos** | |
| 9 | T: Fundamentos de herencia en Java; la palabra clave extends; creación de clases base y derivadas.  P: Creación de jerarquía de clases con herencia; implementación de métodos sobrescritos en clases derivadas. | |
| 10 | T: Concepto de polimorfismo en Java; clases y métodos abstractos; la palabra clave abstract.  P: Desarrollo de programas que demuestren polimorfismo; creación y uso de clases y métodos abstractos. | |
| 11 | T: Interfaces en Java; comparación entre interfaces y clases abstractas; implementación múltiple de interfaces.  P: Implementación de interfaces en clases; creación de programas que utilicen múltiples interfaces. | |
| 12 | T: Sobrecarga de métodos (mismo nombre, diferentes parámetros) y sobreescritura de métodos (redefinición de métodos en clases derivadas).  P: Ejercicios de sobrecarga y sobreescritura de métodos; desarrollo de aplicaciones que utilicen ambos conceptos.  Creación de un sistema de gestión de empleados con jerarquías y métodos específicos. | |
| **Producto o evidencia de aprendizaje:**  Creación de jerarquías de clases que demuestren la herencia, la implementación de métodos sobrescritos en clases derivadas, la realización de ejercicios que utilicen polimorfismo mediante interfaces y clases abstractas, Avance del Informe Académico | | |
| **Instrumentos de evaluación del logro de aprendizaje de la unidad:** | | * Rúbrica. * Informe Académico * Resolución de casos prácticos. |

|  |  |  |
| --- | --- | --- |
| **UNIDAD DIDÁCTICA IV: Persistencia**  **Logro o resultado de aprendizaje de la unidad:**  Al finalizar esta unidad, el estudiante será capaz de gestionar el almacenamiento de datos en aplicaciones Java utilizando técnicas de persistencia. Será capaz de conectar sus aplicaciones a bases de datos mediante JDBC, realizar operaciones de manipulación de datos (consultas, inserciones, actualizaciones y eliminaciones), y manejar excepciones para asegurar la robustez y continuidad del programa durante la interacción con bases de datos. | | |
| **Semana** | **Contenidos Teórico-Prácticos** | |
| 13 | T: Introducción a la persistencia de datos; JDBC (Java Database Connectivity); Conexión a bases de datos, manejo de ResultSet.  P: Conexión a bases de datos MySQL; ejecución de consultas básicas (SELECT, INSERT, UPDATE, DELETE) usando JDBC; | |
| 14 | T: Introducción a ORM (Object-Relational Mapping); Hibernate como herramienta ORM; Hibernate en proyectos Java.  P: Integración de Hibernate en proyectos Java; mapeo de clases a tablas de base de datos; operaciones CRUD usando Hibernate. | |
| 15 | T: Gestión de transacciones en bases de datos; uso de transacciones en JDBC.  P: Implementación de transacciones con JDBC y Hibernate, aplicaciones que manejen múltiples operaciones en una transacción.  Presentación y Exposición del Informe Académico. | |
| 16 | **EXAMEN FINAL** | |
| **Producto o evidencia de aprendizaje:**  implementación de programas que utilicen JDBC para conectarse a bases de datos, operaciones CRUD (Crear, Leer, Actualizar, Borrar) sobre una base de datos, y la documentación detallada del código que maneja la persistencia de datos, asegurando su correcta funcionalidad.  Final de Informe Académico | | |
| **Instrumentos de evaluación del logro de aprendizaje de la unidad:** | | * Rúbrica. * Resolución de casos prácticos. * Cuestionario * Trabajo monográfico. * Exposición de trabajos |

# ESTRATEGIAS DIDÁCTICAS

Se considera para el cumplimiento y desarrollo de las competencias generales, específicas y los logros de aprendizaje las siguientes estrategias de trabajo:

* El docente pondrá, según calendario académico, a disposición de los estudiantes, materiales de clase, lecturas complementarias y otros materiales relevantes en el aula virtual del curso para disponibilidad de los estudiantes.
* Los estudiantes tendrán acceso al aula virtual donde visualizarán materiales de clase, videos y enlaces a lecturas complementarias a los temas de cada semana para comentarlos participativamente en los foros y en las sesiones de clase. Asimismo, los estudiantes encontrarán actividades y autoevaluaciones que les permitirá reforzar su proceso de aprendizaje.
* Los estudiantes tendrán acceso al sistema de información ERP EDUCA, en donde podrán visualizar la información académica del curso: horario, ambiente, docente, notas y otros.
* El docente hará un seguimiento de las actividades configuradas en el aula virtual como foros, tareas, autoevaluaciones a fin de que el estudiante participe activa y colaborativamente con sus compañeros de clase. Resolución de ejercicios y problemas.
* Desarrollo de experiencias prácticas en laboratorio con guías de desarrollo.

**Uso de fuentes académicas y citado**

Los estudiantes elaborarán durante el curso un informe académico original, el cual deberá utilizar correctamente fuentes académicas y citado. Para ello:

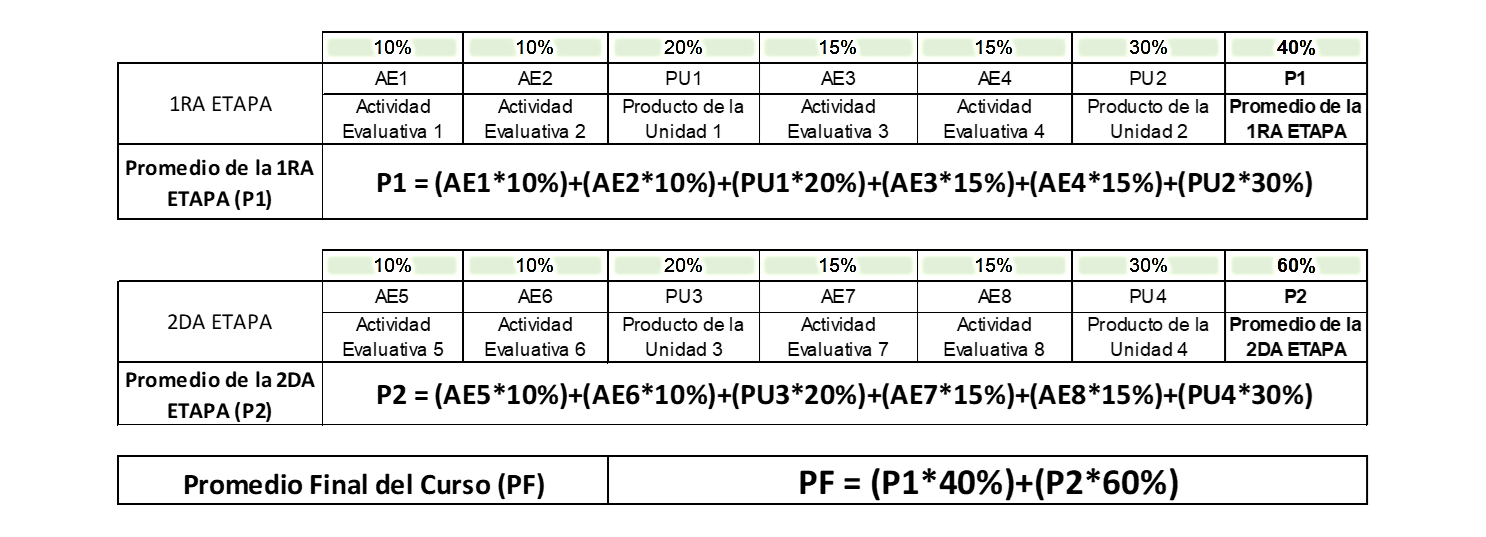
* El docente entregará a los estudiantes un tema específico relacionado con el curso. Este tema será desarrollado según la estructura propuesta en el documento de proyecto final generado para el curso.
* Los avances entregados por los estudiantes deberán mantener la originalidad, para lo cual se utilizará la herramienta Turnitin.

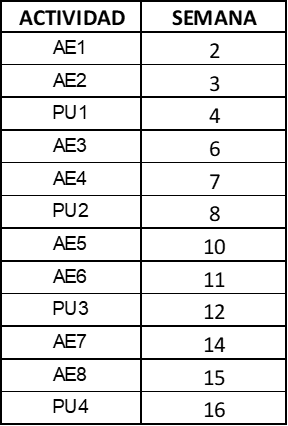
**Exposiciones y presentaciones académicas**

Los estudiantes realizarán exposiciones de los avances periódicos de sus trabajos de investigación. Durante estas presentaciones, la audiencia participará activamente en la elaboración de consultas y comentarios mediados por el docente.

# SISTEMA DE EVALUACIÓN DEL APRENDIZAJE

La evaluación es formativa y permanente y se deben centrar en los logros o resultados del aprendizaje que a su vez se dirigen a consolidar adquisiciones del perfil de egreso. Las evidencias del logro adquirido se presentarán a través de las siguientes actividades:

****

****

**"Este curso no tiene Examen Sustitutorio"**

# MEDIOS Y MATERIALES EDUCATIVOS

El curso contiene medios y materiales educativos especialmente diseñados para la interacción entre los estudiantes y docentes en actividades de aprendizaje en el aula de clase, en el laboratorio de computación, y como apoyo a través del aula virtual y un sistema multimedia que aseguran la interacción con los materiales y diversos medios sociales y coadyuvan en el logro de aprendizajes.

**Medios**:

* Aula Virtual (Blackboard Learn Ultra)
* ERP Educa, MySQLWorkBench
* Padlet
* Mentimeter
* software DIA
* Erwin Data Modeler
* plataformas online para la elaboración de diseño conceptual y lógicos

**Materiales:**

* Sílabo
* PPT
* Material Multimedia
* Libros digitales
* Guía de laboratorio
* Formato de Informe Académico

# BIBLIOGRAFÍA

Horstmann, C. S. (2020). Core Java Volume I: Fundamentals (11ª ed.). Prentice Hall. https://www.pearson.com/us/high

Bloch, J. (2018). Effective Java (3ª ed.). Addison-Wesley. https://www.pearson.com/store/p/effective-java/P100000256245

Lafore, R. (2002). Data Structures and Algorithms in Java (3ª ed.). Sams Publishing. https://www.pearson.com/store/p/data-structures-and-algorithms-in-java/P100000209388

Sierra, K., & Bates, B. (2005). Head First Java (2ª ed.). O'Reilly Media. https://www.oreilly.com/library/view/head-first-java/0596009208/

Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley. https://www.pearson.com/store/p/patterns-of-enterprise-application-architecture/P100000035282

Gosling, J., Joy, B., Steele, G., & Bracha, G. (2005). The Java Programming Language (5ª ed.). Addison-Wesley. https://www.pearson.com/store/p/java-programming-language-the/P100000655946

Schildt, H. (2019). Java: A Beginner's Guide (8ª ed.). McGraw-Hill Education. https://www.mhprofessional.com/9781260440232-usa-java-beginners-guide-8th-edition

Stevens, C., & Beazley, R. (2018). Java Persistence with Hibernate. Manning Publications. https://www.manning.com/books/java-persistence-with-hibernate

Oracle. (n.d.). Documentación Oficial de Java. Retrieved July 27, 2024, from <https://docs.oracle.com/javase/>

Oracle. (n.d.). Tutoriales y Recursos de Oracle. Retrieved July 27, 2024, from <https://www.oracle.com/java/technologies/javase-tutorials.html>

JournalDev. (n.d.). Java Design Patterns. Retrieved July 27, 2024, from https://www.journaldev.com/java-design-patterns