

Curso: Programación Estructurada		Horas aula: 3
Clave: 061CP040		Horas plataforma: 1
Antecedentes:		Horas laboratorio: 1
Competencia del área: Desarrollar software y servicios de soporte técnico y redes, con la finalidad de solucionar problemas y agilizar procesos en la toma de decisiones en empresas públicas y privadas, bajo estándares de calidad nacional e internacional, a través del análisis de problemas, comunicación, liderazgo e innovación.	Competencia del curso: Desarrollar programas mediante el uso lenguajes con un enfoque estructurado, en un entorno de desarrollo integrado, con el fin de dar solución a los problemas y servir de apoyo en la toma de decisiones dentro de las organizaciones, aplicando los principios de una buena programación poniendo en práctica el análisis de problemas, el trabajo en equipo y la responsabilidad.	
Elementos de competencia:		
<ol style="list-style-type: none"> 1. Elaborar programas sencillos, en un entorno de desarrollo integrado, con el fin de sentar las bases para la solución de problemas más complejos, aplicando los principios de una buena programación y poniendo en práctica las habilidades de análisis de problemas, el trabajo en equipo y la responsabilidad. 2. Construir programas aplicando estructuras de control selectivas y repetitivas, en un entorno de desarrollo integrado con el fin de ofrecer soluciones eficientes a problemas más complejos, aplicando los principios de una buena programación y poniendo en práctica las habilidades de análisis de problemas, el trabajo en equipo y la responsabilidad. 3. Desarrollar programas en un entorno de desarrollo integrado, usando arreglos, como una solución a problemas de representación de datos que surgen en las organizaciones y diseñar propuestas donde se apliquen los métodos de clase, los principios de una buena programación y las habilidades de análisis de problemas, el trabajo en equipo y la responsabilidad. 		
Perfil del docente:		
Ingeniero en Software, Licenciado en Sistemas Computacionales Administrativos, Licenciado en Informática. Preferentemente con Maestría en Ciencias Computacionales, Ingeniería en Software o Sistemas Computacionales. Además, debe de contar con experiencia docente en enfoque por competencias, centrados en el estudiante y en la construcción de ambientes de aprendizajes autónomos y colaborativos en donde se incorporen el uso de las TIC.		
Elaboró: LORENIA CANTU BALLESTEROS		Marzo 2021
Revisó: ALMA ISABEL ARIAS HURTADO		Abril 2021
Última actualización:		
Autorizó: Coordinación de Procesos Educativos		

Elemento de competencia 1: Elaborar programas sencillos, en un entorno de desarrollo integrado, con el fin de sentar las bases para la solución de problemas más complejos, aplicando los principios de una buena programación y poniendo en práctica las habilidades de análisis de problemas, el trabajo en equipo y la responsabilidad.

Competencias blandas a promover: Análisis de problemas, trabajo en equipo y responsabilidad.

EC1 Fase I: Introducción a los lenguajes de programación.

Contenido: Historia de los lenguajes de programación, paradigmas de programación y lenguaje de programación Java y sus características.

EC1 F1 Actividad de aprendizaje 1: Línea del tiempo sobre la historia de los lenguajes de programación.

Elaborar en equipo una línea del tiempo sobre los lenguajes de programación a partir de la revisión del video sugerido en los recursos y la consulta de fuentes confiables en internet para considerar los lenguajes surgidos de 2005 a la fecha que no están incluidos en el video.

Tomar las notas necesarias en el cuaderno, previo a la clase, para desarrollar el esquema y construir la línea del tiempo en el aula con la información obtenida, utilizando Word o alguna otra herramienta digital de preferencia.

Integrar documento de acuerdo a las consideraciones del facilitador y entregar según lo indique.

1 hr. Aula
1 hr. Plataforma

Tipo de actividad:

Aula (X) Plataforma (X) Laboratorio ()
Grupal () Individual () Equipo (X)
Independiente ()

Recursos:

- Video: [Línea de tiempo lenguajes de programación](#)
- Aplicación: Word o aplicación de acceso libre en Internet
- Internet

Criterios de evaluación de la actividad:

- Rúbrica: [Línea del tiempo](#)

EC1 F1 Actividad de aprendizaje 2: Esquema gráfico: tabla sobre los paradigmas de programación.

Desarrollar en equipo una tabla de cinco entradas sobre los paradigmas de programación con las siguientes etiquetas de columnas: paradigma (nombre del paradigma), características, ventajas, desventajas y lenguajes de programación (ejemplos de lenguajes que apliquen al paradigma).

Incluir en la tabla información de los siguientes paradigmas de programación:

- a) imperativo,
- b) estructurado,
- c) declarativo: funcional y lógico,
- d) orientado a objetos,
- e) de programación por eventos, y
- f) paralelo distribuido y concurrente.

Consultar para el desarrollo de la tabla, el capítulo

Tipo de actividad:

Aula (X) Plataforma () Laboratorio (X)
Grupal () Individual () Equipo (X)
Independiente ()

Recursos:

- [Introducción a la programación](#). Juganaru Mathieu, M.
- Aplicación: Word o una aplicación de acceso libre en Internet

Criterios de evaluación de la actividad:

- Cumplir con los criterios especificados para la tabla (etiquetas): paradigma, características, ventajas, desventajas y lenguajes de programación.
- Incluir todos los temas solicitados.
- Atender la redacción, ortografía, citas y referencias en formato APA.

<p>1 del libro señalado en los recursos y concretar en el laboratorio, un documento bajo los criterios señalados por el facilitador.</p> <p>2 hrs. Aula 1 hr. Laboratorio</p>	
<p>EC1 F1 Actividad de aprendizaje 3: Mapa de cajas sobre la historia y características del lenguaje de programación Java.</p> <p>Desarrollar en equipo un mapa de cajas que sintetice los aspectos más relevantes sobre la historia del lenguaje programación Java y sus características;partiendo de la consulta de fuentes confiables y tomando como base la consulta del video señalado en el apartado de recursos.</p> <p>Incluir de la historia de Java al menos: ¿Quién fue su creador?, ¿Qué empresa lo patrocinó?, ¿En qué lenguaje se basó?, ¿Cuál fue su nombre inicial?, ¿En qué año surgió? y versiones del <i>Java development kit</i> (JDK).</p> <p>Incluir y describir las siguientes características de Java: orientado a objetos, sencillo, distribuido, portable, interpretado y compilado, robusto, seguro, independiente de la arquitectura, alto rendimiento, multihilo y dinámico.</p> <p>2 hrs. Plataforma</p>	<p>Tipo de actividad: Aula () Plataforma (X) Laboratorio () Grupal () Individual () Equipo (X) Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Video: Mapa de caja • Aplicación: Word o una aplicación de acceso libre en Internet • Internet y/o eLibro <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Cumplir con los criterios del esquema: Tres niveles con cajas: tema central en el primer nivel, los subtemas en el segundo nivel y en el tercer nivel se sintetiza una descripción de cada subtema. • Incluir todos los temas solicitados.
<p>EC1 Fase II: Fundamentos para desarrollar programas en Java.</p> <p>Contenido: Estructura de un programa en Java (import, class y package, main, palabras reservadas, comentarios y signos de puntuación), elementos de un programa en Java (tipos de datos primitivos, variables, constantes, contadores, acumuladores y operadores), principios de una buena programación y convenciones de código en Java, entrada y salida de datos.</p>	
<p>EC1 F2 Actividad de aprendizaje 4: Apuntes de clase sobre la estructura de un programa en Java.</p> <p>Desarrollar en equipo y en aula, apuntes de clase de manera digital sobre la estructura de un programa en Java.</p> <p>Organizar la información mediante diversos esquemas (tablas, cuadros sinópticos, mapas conceptuales, etc.), listas numeradas y/o imágenes.</p> <p>Incluir en los apuntes digitales la siguiente información:</p> <ul style="list-style-type: none"> • a) package, import, class, main (definirlos e 	<p>Tipo de actividad: Aula (X) Plataforma () Laboratorio () Grupal (X) Individual () Equipo (X) Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Java 2: Lenguaje y aplicaciones. Ceballos Sierra, F. J. • Aplicación: Word o una aplicación de acceso libre en Internet • Internet y/o eLibro <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Rúbrica de apuntes de clase • Citas y referencias en formato APA 7

<p>identificarlos en un programa),</p> <ul style="list-style-type: none"> • b) paquetes más importantes (con una breve descripción), • c) palabras reservadas (qué son y algunos ejemplos), • d) comentarios (para qué sirven y los signos utilizados), • e) signos de puntuación (para qué sirven y la lista de signos). <p>Revisar, previo a la clase, el capítulo 5 del libro recomendado en los recursos y tener ubicadas otras fuentes confiables en eLibro e Internet, para concretar los apuntes en la clase.</p> <p>Concluir el tema mediante sesión de preguntas y respuestas que realizará el profesor para aclarar y/o reforzar los temas.</p> <p>2 hrs. Aula</p>	
<p>EC1 F2 Actividad de aprendizaje 5: Infografía sobre los elementos de un programa en Java.</p> <p>Desarrollar en equipo, una infografía sobre los elementos de un programa en Java incorporando la siguiente información:</p> <ul style="list-style-type: none"> • a) Tipos de datos primitivos: Cuáles son, tamaño en bytes y rango de valores de cada uno de ellos, • b) definir una variable y cómo se declara en Java, • c) definir una constante y cómo se declara en Java, • d) definir contador y acumulador, • e) operadores aritméticos, relacionales, lógicos, de incremento y decremento, combinados, precedencia de los operadores: lista de operadores y un ejemplo. <p>Revisar el capítulo 1 del libro recomendado en los recursos y otras fuentes confiables y organizar previo a la clase, la información en cada uno de los esquemas seleccionados para concretar la infografía en el laboratorio.</p> <p>Presentar la infografía para recibir retroalimentación por parte del facilitador.</p> <p>1 hr. Aula 1 hr. Laboratorio</p>	<p>Tipo de actividad: Aula (X) Plataforma () Laboratorio (X) Grupal (X) Individual () Equipo (X) Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Sitio web para diseño gráfico: Canva • Libro: Programación orientada a objetos en Java. Blasco, F. • Aplicación: Word o una aplicación de acceso libre en Internet <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Rúbrica: Infografía
<p>EC1 F2 Actividad de aprendizaje 6: Foro: mejorar la calidad de la programación en Java.</p>	<p>Tipo de actividad: Aula () Plataforma (X) Laboratorio ()</p>

<p>Participar en el foro denominado: mejorar la calidad de la programación en Java. Considerando los siguientes aspectos:</p> <ul style="list-style-type: none"> • Revisar a detalle el documento sobre las convenciones en Java, que se coloca en el apartado de recursos así como el apéndice B sobre convenios de programación en Java, del libro que se ubica en la misma sección. • Buscar en Internet la propuesta de Christopher Diggins, sobre los 18 principios de una buena programación: DRY (Don't repeat yourself), abstraction principle, KISS (Keep it simple, stupid), avoid creating a YAGNI, do the simplest thing that could possibly work, don't make me think, open/closed principle, write code for the maintainer, principle of least astonishment, single responsibility principle, minimize coupling, maximize cohesion, hide implementation details, law of demeter, avoid premature optimization, code reuse is good, separation of concerns and embrace change y revisar a detalle la información. • Comentar en el foro, al menos una convención en Java o un principio de una buena programación, con una extensión mínima de 60 palabras. • Demostrar buena redacción y ortografía. • Describir al final de la participación, la fuente de información consultada en formato APA. • Participar, al menos una vez, realimentando el comentario de un compañero. <p>2 hrs. Plataforma</p>	<p>Grupal () Individual (X) Equipo () Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Documento en web: Convención de código fuente • Libro: Programación en Java 2. Sánchez Allende, J. • Internet <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Rúbrica: Participación en foro
<p>EC1 F2 Actividad de aprendizaje 7: Tabla de triple entrada sobre clases y métodos para entrada y salida de datos en Java.</p> <p>Desarrollar en equipo, una tabla de triple entrada sobre las clases y métodos que utiliza Java para la entrada y salida de datos utilizando las siguientes etiquetas de columnas: clase/método, descripción y sintaxis y/o ejemplo (segmento de código en Java).</p> <p>Incluir en la tabla, las clases y/o métodos:</p> <ul style="list-style-type: none"> • Scanner, • System.out.print, • System.out.println, • nextByte(), • nextShort(), • nextInt(), • nextLong(), • nextFloat(), • nextDouble(), • next() • nextLine(). 	<p>Tipo de actividad:</p> <p>Aula () Plataforma (X) Laboratorio ()</p> <p>Grupal () Individual () Equipo (X) Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Libro: Empezar a programar usando java. Prieto, N. - • Aplicación: Word o una aplicación de acceso libre en Internet • Internet y/o eLibro <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Cumplir con los criterios especificados para la tabla (etiquetas): clase/método, descripción y sintaxis y/o ejemplo. • Abarcar todas las clases y/o métodos solicitados. • Atender la redacción, ortografía, citas y referencias

<p>Revisar el capítulo 7, del libro recomendado en los recursos, y apoyarse en otras fuentes confiables en Internet o eBook para el desarrollo correcto de la actividad.</p> <p>2 hrs. Plataforma</p>	<p>en formato APA.</p>
<p>EC1 F2 Actividad de aprendizaje 8: Programas en Java para entrada y salida de datos por consola.</p> <p>Elaborar programas en Java, para dar solución a los problemas planteados, utilizando los métodos de entrada y salida de datos por consola. Tanto en aula, como en el laboratorio, se deberá trabajar en binas de estudiantes, apoyados por el facilitador.</p> <ul style="list-style-type: none"> • Utilizar un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para elaborar los programas, tal como Netbeans. • Revisar el enlace a la página oficial de descargas del IDE de Netbeans, que se muestra en los recursos, para instalarlo en las computadoras personales. • Ver el video el tutorial y consultar el capítulo 7, que se presentan como apoyo a la actividad en el apartado de recursos. • Entregar los programas, en una tabla de doble entrada, donde se incluya: el código fuente a partir del método main(), así como, la captura de pantalla del código fuente completo y la salida a consola. Cada tabla tendrá como encabezado el enunciado del problema y será una para cada programa. • Integrar un documento que incluya una portada y las tablas de cada programa. Subir a la plataforma. <p>3 hrs. Aula 2 hrs. Laboratorio</p>	<p>Tipo de actividad: Aula (X) Plataforma () Laboratorio (X) Grupal () Individual () Equipo (X) Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Listado de programas para entrada y salida de datos por consola. • Página web: Oracle/descargas/Netbeans • Video: Programación en Java, entrada y salida de datos por consola • IDE: Netbeans • Libro: Empezar a programar usando Java. Prieto, N. <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Cumplir con los criterios que se indican en la tabla de doble entrada (etiquetas): código fuente (primera columna) y captura de pantalla del código fuente completo y salida a consola (segunda columna). • Entregar el total de programas solicitados. • Entregar programas codificados: Sin errores de sintaxis. Sin errores lógicos. Con todos los requerimientos específicos de entrada y salida de datos. Sin errores ortográficos en el código fuente. • Sin errores ortográficos en el documento integrado.
<p>Evaluación formativa:</p> <ul style="list-style-type: none"> • Línea del tiempo sobre la historia de los lenguajes de programación. • Esquema gráfico: tabla sobre los paradigmas de programación. • Mapa de cajas sobre la historia y características del lenguaje de programación Java. • Apuntes de clase sobre la estructura de un programa en Java. • Infografía sobre los elementos de un programa en Java. • Foro: mejorar la calidad de la programación en Java. • Tabla de triple entrada sobre clases y métodos para entrada y salida de datos en Java. • Programas en Java para entrada y salida de datos por consola. 	
<p>Fuentes de información</p>	

1. Álvarez Valdeón, I. (19 de diciembre de 2011). *Convención de código fuente*. https://joinup.ec.europa.eu/sites/default/files/document/2012-01/ConvencionCodigo_JOINUP_20111219_1.0.pdf
2. Blasco, F. (2019). Programación orientada a objetos en Java. Ediciones de la U. <https://elibro.net/es/ereader/ues/127125?page=15>
3. Canva (03 de marzo de 2021). https://www.canva.com/es_mx/
4. Ceballos Sierra, F. J. (2015). *Java 2: lenguaje y aplicaciones*. RA-MA Editorial. <https://elibro.net/es/lc/ues/titulos/62458>
5. Deitel, P. (2016). *Cómo programar en Java*. Pearson.
6. Joyanes Aguilar, L. (2011). *Programación en Java 6. Algoritmos y programación orientada a objetos*. McGraw-Hill.
7. Joyanes Aguilar, L. (2013). *Fundamentos generales de programación*. McGrawHill.
8. Juganaru Mathieu, M. (2015). *Introducción a la programación*. Grupo Editorial Patria. <https://elibro.net/es/ereader/ues/39449?page=16>
9. Oracle. (03 de marzo de 2021). *Descargas*. <https://www.oracle.com/technetwork/es/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html>
10. Prieto, N. (2013). *Empezar a programar usando java*. Editorial de la Universidad Politécnica de Valencia. <https://elibro.net/es/ereader/ues/54085?page=167>
11. Profe Dam. (06 de julio de 2020). *Mapa de cajas* [Video]. YouTube. <https://youtu.be/cBwej5Fr63E>
12. Programación ATS. (6 de febrero de 2016). *Programación en Java || Introducción || Entrada y salida de datos por consola* [Video]. YouTube. <https://www.youtube.com/watch?v=taSF09mPlcs>
13. Sánchez Allende, J. (2005). *Programación en Java 2*. McGraw-Hill España. <https://elibro.net/es/ereader/ues/50077?page=416>
14. Sznajdleder, P. (2020). *Java a fondo*. Alfaomega.
15. Trejos Buriticá, O. I. (2017). *Lógica de programación*. Ediciones de la U. <https://elibro.net/es/lc/ues/titulos/70315>.
16. Vargas Martínez, S. (30 de marzo de 2016). *Línea de tiempo lenguajes de programación* [Video]. YouTube. <https://youtu.be/VQ7--DqIMGw>

Elemento de competencia 2: Construir programas aplicando estructuras de control selectivas y repetitivas, en un entorno de desarrollo integrado con el fin de ofrecer soluciones eficientes a problemas más complejos, aplicando los principios de una buena programación y poniendo en práctica las habilidades de análisis de problemas, el trabajo en equipo y la responsabilidad.

Competencias blandas a promover: Análisis de problemas, trabajo en equipo y responsabilidad.

EC2 Fase I: Estructuras de control selectivas

Contenido: if...else y switch.

EC2 F1 Actividad de aprendizaje 9: Programas en Java aplicando la estructura selectiva if...else.

Elaborar programas en Java aplicando la estructura selectiva if...else, para dar solución a la lista de problemas planteados por el facilitador.

- Consultar y tomar notas en el cuaderno, previo a la clase, sobre la sintaxis del if...else, apoyándose en el capítulo 7 del libro sugerido en los recursos, así como, en los dos videos tutoriales que aparecen en la misma sección. De ser necesario, puede consultar alguna fuente confiable adicional.
- Utilizar en el aula, el IDE de Netbeans, para codificar los programas en Java, respetando los principios de una buena programación.
- Entregar la sintaxis del if...else y los programas, los cuales estarán organizados en una tabla de doble entrada que incluya: el código fuente a partir del método main(), la captura de pantalla del código fuente completo y la salida a consola. Cada tabla tendrá como encabezado el enunciado del problema y será una por programa.
- Integrar un documento en Word, o en una aplicación de acceso libre en Internet, con una portada, la sintaxis del if...else, las tablas de cada programa y las fuentes consultadas en formato APA.
- Convertir a PDF el documento y subirlo a plataforma.

3 hrs. Aula
2 hrs. Plataforma

Tipo de actividad:

Aula (X) Plataforma (X) Laboratorio ()
Grupal (X) Individual (X) Equipo ()
Independiente ()

Recursos:

- Listado de programas aplicando la estructura selectiva if...else
- Video: [Curso Java - 3: Sentencias de control 1 \(if – else, switch\)](#)
- Video: [Tutorial 7 de Java - Estructuras if...else & if anidados](#)
- Aplicación: Word o aplicación de acceso libre en Internet
- IDE: Netbeans
- Libro: [Java 2. Lenguaje y aplicaciones](#). Ceballos Sierra, F. J.

Criterios de evaluación de la actividad:

- Cumplir con los criterios que se indican en la tabla de doble entrada (etiquetas): código fuente (primera columna) y captura de pantalla del código fuente completo y salida a consola (segunda columna).
- Entregar el total de los programas solicitados.
- Entregar el total de los programas codificados: Sin errores de sintaxis. Sin errores lógicos. Con todos los requerimientos específicos de entrada y salida de datos. Sin errores ortográficos en el código fuente.
- Sin errores ortográficos en el documento integrado.

EC2 F1 Actividad de aprendizaje 10: Práctica de programación en Java aplicando la estructura selectiva switch.

Desarrollar la práctica de programación en Java que proporcione el facilitador, aplicando la estructura selectiva switch, a la solución de problemas planteados.

- Consultar y tomar notas en el cuaderno, previo a la práctica, sobre la sintaxis de la estructura switch, apóyese en el capítulo 7 del

Tipo de actividad:

Aula (X) Plataforma () Laboratorio (X)
Grupal (X) Individual (X) Equipo ()
Independiente ()

Recursos:

- Práctica de programación aplicando la estructura selectiva switch
- Video: [Curso Java - 3: Sentencias de control 1 \(if – else, switch\)](#)
- Aplicación: Word o aplicación de acceso libre en

<p>libro que se presenta en los recursos, así como, en el video tutorial que se muestra en la misma sección. De ser necesario, puede consultar alguna fuente confiable adicional y realizarlo en el aula y/o antes de la sesión presencial.</p> <ul style="list-style-type: none"> • Concretar el tema en el aula y avanzar en la resolución de los problemas planteados. • Utilizar en el laboratorio el IDE de Netbeans, para codificar en Java la solución a los problemas planteados en la práctica, respetando los principios de una buena programación. • Entregar la sintaxis del switch y los programas de la práctica de programación, los cuales estarán organizados en una tabla de doble entrada que incluya: el código fuente a partir del método main(), la captura de pantalla del código fuente completo y la salida a consola. Cada tabla tendrá como encabezado el enunciado del problema y será una para cada programa. • Integrar un documento en Word, o en una aplicación de acceso libre en Internet, con una portada, la sintaxis del switch, las tablas de cada programa de la práctica de programación y las fuentes consultadas en formato APA. • Convertir a PDF el documento y subirlo a plataforma. <p>3 hrs. Aula 2 hrs. Laboratorio</p>	<p>Internet</p> <ul style="list-style-type: none"> • IDE: Netbeans • Libro: Java 2: lenguaje y aplicaciones. Ceballos Sierra, F. J. <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Cumplir con los criterios que se indican en la tabla de doble entrada (etiquetas): código fuente (primera columna) y captura de pantalla del código fuente completo y salida a consola (segunda columna). • Entregar el total de los programas solicitados. • Entregar el total de los programas codificados: Sin errores de sintaxis. Sin errores lógicos. Con todos los requerimientos específicos de entrada y salida de datos. Sin errores ortográficos en el código fuente. • Sin errores ortográficos en el documento integrado.
--	--

EC2 Fase II: Estructuras de control repetitivas

Contenido: For, while y do while

EC2 F2 Actividad de aprendizaje 11: Quiz sobre las estructuras de control repetitivas for, while y do while

Contestar el quiz incluido en la plataforma, sobre las estructuras de control repetitivas (ciclos o bucles) for, while y do while. Para lo cual deberá:

- Elaborar apuntes digitales o en cuaderno sobre los siguientes temas: Sintaxis de las estructuras de control repetitivas: for, while y do while. Definición de cada una de las estructuras repetitivas: for, while y do while. Identificar las características y/o elementos de los ciclos while y do while controlados por contador, centinela, preguntas al usuario y banderas.
- Consultar el capítulo 7, del libro sugerido en los recursos, y al menos uno más de la lista de fuentes de información que se proporcionan en la secuencia didáctica, con el

Tipo de actividad:

Aula () Plataforma (X) Laboratorio ()
 Grupal () Individual (X) Equipo ()
 Independiente ()

Recursos:

- Libro: [Java 2: lenguaje y aplicaciones](#). Ceballos Sierra, F. J.
- Fuentes de información de la secuencia didáctica

Criterios de evaluación de la actividad:

- Sumatoria de los puntos obtenidos en las preguntas respondidas correctamente en el quiz (escala del 0 al 10).

<p>fin de concretar los apuntes.</p> <ul style="list-style-type: none"> • Estudiar los apuntes generados. • Acceder al quiz disponible en la plataforma institucional y responderlo en un máximo de dos oportunidades, en la fecha y tiempo establecido (30 minutos). • Avanzar hasta concluir con todas las preguntas y enviar para su revisión. <p>2 hrs. Plataforma</p>	
<p>EC2 F2 Actividad de aprendizaje 12: Programas en Java aplicando la estructura repetitiva for.</p> <p>Elaborar programas en Java aplicando la estructura repetitiva for, para dar solución a los problemas planteados por el facilitador.</p> <ul style="list-style-type: none"> • Revisar en el aula el video que se muestra en la sección de recursos sobre el ciclo for y concretar el tema, avanzando en el resolución de los problemas planteados. • Utilizar en el laboratorio el IDE de Netbeans para codificar los programas, respetando los principios de una buena programación. • Entregar los programas, los cuales estarán organizados en una tabla de doble entrada que incluya: el código fuente a partir del método main(), la captura de pantalla del código fuente completo y la salida a consola. Cada tabla tendrá como encabezado el enunciado del problema y será una para cada programa. • Integrar un documento en Word, o en una aplicación de acceso libre en Internet, con una portada, las tablas de cada programa y las fuentes consultadas en formato APA. <p>6 hrs. Aula 1 hr. Laboratorio</p>	<p>Tipo de actividad: Aula (X) Plataforma () Laboratorio (X) Grupal (X) Individual (X) Equipo () Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Listado de programas para aplicar la estructura repetitiva for • Video: Programación en Java. Ciclo for • Aplicación: Word o aplicación de acceso libre en Internet • IDE: Netbeans <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Cumplir con los criterios que se indican en la tabla de doble entrada (etiquetas): código fuente (primera columna) y captura de pantalla del código fuente completo y salida a consola (segunda columna). • Entregar el total de programas solicitados. Entregar programas codificados: Sin errores de sintaxis. Sin errores lógicos. Con todos los requerimientos específicos de entrada y salida de datos. Sin errores ortográficos en el código fuente. • Sin errores ortográficos en el documento integrado.
<p>EC2 F2 Actividad de aprendizaje 13: Ejercicios en Java aplicando la estructura repetitiva while.</p> <ul style="list-style-type: none"> • Construir programas en Java aplicando la estructura repetitiva while, para dar solución a los problemas planteados, en los ejercicios que proporcione el facilitador. • Revisar en el aula el video que se muestra en la sección de recursos sobre el ciclo while y concretar el tema, avanzando en el resolución de los problemas planteados. • Utilizar en el laboratorio el IDE de Netbeans, para codificar los programas en Java, respetando los principios de una buena programación. • Entregar los programas, los cuales estarán 	<p>Tipo de actividad: Aula (X) Plataforma () Laboratorio (X) Grupal (X) Individual (X) Equipo () Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Listado de ejercicios aplicando la estructura repetitiva while • Video: Programación en Java. Ciclo while • Aplicación: Word o aplicación de acceso libre en Internet • IDE: Netbeans <p>Criterios de evaluación de la actividad:</p>

<p>organizados en una tabla de doble entrada que incluya: el código fuente a partir del método main(), la captura de pantalla del código fuente completo y la salida a consola. Cada tabla tendrá como encabezado el enunciado del problema y será una para cada programa entregado.</p> <ul style="list-style-type: none"> • Integrar un documento en Word, o en una aplicación de acceso libre en Internet, con una portada, las tablas de cada programa y las fuentes consultadas en formato APA. Subirlo documento a plataforma. <p>6 hrs. Aula 2 hrs. Laboratorio</p>	<ul style="list-style-type: none"> • Cumplir con los criterios que se indican en la tabla de doble entrada: código fuente (primera columna) y captura de pantalla del código fuente completo y salida a consola (segunda columna). • Entregar el total de los programas solicitados. • Entregar el total de los programas codificados: Sin errores de sintaxis. Sin errores lógicos. Con todos los requerimientos específicos de entrada y salida de datos. Sin errores ortográficos en el código fuente. • Sin errores ortográficos en el documento integrado.
<p>EC2 F2 Actividad de aprendizaje 14: Práctica de programación en Java aplicando la estructura repetitiva do while</p> <ul style="list-style-type: none"> • Desarrollar en binas la práctica de programación en Java, que proporcione el facilitador, aplicando la estructura repetitiva do while a la solución de problemas planteados. • Revisar en el aula el video que se muestra en la sección de recursos sobre el ciclo do while y concretar el tema avanzando en la resolución de los problemas planteados. • Utilizar en el laboratorio, el IDE de Netbeans para codificar en Java la solución a los problemas planteados en la práctica, respetando los principios de una buena programación. • Entregar los programas de la práctica de programación, los cuales estarán organizados en una tabla de doble entrada que incluya: el código fuente a partir del método main(), la captura de pantalla del código fuente completo y la salida a consola. Cada tabla tendrá como encabezado el enunciado del problema y será una para cada programa. • Integrar un documento en Word, o en una aplicación de acceso libre en Internet, con una portada, las tablas de cada programa de la práctica de programación y las fuentes consultadas en formato APA. <p>6 hrs. Aula 2 hrs. Laboratorio</p>	<p>Tipo de actividad: Aula (X) Plataforma () Laboratorio (X) Grupal (X) Individual () Equipo (X) Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Práctica de programación aplicando la estructura repetitiva do while • Video: Programación en Java. Ciclo do while • Aplicación: Word o aplicación de distribución libre en Internet • IDE: Netbeans <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Cumplir con los criterios que se indican en la tabla de doble entrada (etiquetas): código fuente (primera columna) y captura de pantalla del código fuente completo y salida a consola (segunda columna). • Entregar el total de los programas solicitados. • Entregar el total de los programas codificados: Sin errores de sintaxis. Sin errores lógicos. Con todos los requerimientos específicos de entrada y salida de datos. Sin errores ortográficos en el código fuente. • Sin errores ortográficos en el documento integrado.
<p>Evaluación formativa:</p> <ol style="list-style-type: none"> 1. Programas en Java aplicando la estructura selectiva if...else. 2. Práctica de programación en Java aplicando la estructura selectiva switch. 3. Quiz sobre las estructuras de control repetitivas for, while y do while. 4. Programas en Java aplicando la estructura repetitiva for. 	

- 5. Ejercicios en Java aplicando la estructura repetitiva while.
- 6. Práctica de programación en Java aplicando la estructura repetitiva do while.

Fuentes de información

- 8. Programación ATS. (2 de marzo de 2017). 28. *Programación en Java || Ciclos || Ciclo do while* [Video]. YouTube. <https://youtu.be/t2R8ym8hJ4>

Elemento de competencia 3: Desarrollar programas en un entorno de desarrollo integrado, usando arreglos, como una solución a problemas de representación de datos que surgen en las organizaciones y diseñar propuestas donde se apliquen los métodos de clase, los principios de una buena programación y las habilidades de análisis de problemas, el trabajo en equipo y la responsabilidad.

Competencias blandas a promover: Análisis de problemas, trabajo en equipo y responsabilidad.

EC3 Fase I: Arreglos unidimensionales (vectores).

Contenido: Declaración, inicialización, almacenamiento, recuperación y recorrido en arreglos unidimensionales o vectores.

EC3 F1 Actividad de aprendizaje 15: Cuadro de triple entrada sobre la estructura de almacenamiento arreglos en Java.

Desarrollar en equipo una tabla de triple entrada sobre la estructura de almacenamiento arreglos en Java, con las siguientes etiquetas de columnas: tipos de arreglos, características y sintaxis y declaración.

- Incluir en la primera fila, a los arreglos unidimensionales o vectores, y en la segunda, a los arreglos bidimensionales o matrices.
- Consultar, para llenar la tabla, el capítulo 6 del libro que aparece en los recursos, o algún otro incluido en la lista de fuentes de información.
- Utilizar las herramientas de diseño de tabla de Word o alguna otra aplicación de acceso libre disponible en Internet.
- Integrar un documento que incluya una portada, la tabla y las referencias en formato APA. Subirlo a la plataforma para su evaluación

1 hr. Plataforma

Tipo de actividad:

Aula () Plataforma (X) Laboratorio ()
Grupal () Individual () Equipo (X)
Independiente ()

Recursos:

- Aplicación: Word o aplicación de acceso libre en Internet
- Libro: [Java 2: iniciación y referencia](#). Sánchez Allende, J.
- Fuentes de información de la secuencia didáctica

Criterios de evaluación de la actividad:

- Cumplir con los criterios que se indican en la tabla de triple entrada (etiquetas): tipos de arreglos, características y sintaxis y declaración
- Incluir todos los temas solicitados.
- Atender la redacción, ortografía, citas y referencias en formato APA.

EC3 F1 Actividad de aprendizaje 16: Ejercicios en Java usando arreglos unidimensionales (vectores).

Desarrollar programas en Java usando como estructura de almacenamiento un arreglo unidimensional o vector, para dar solución a los problemas planteados en los ejercicios que proporcione el facilitador.

- Revisar previo a la clase, los videos y el capítulo 6 que se proponen en la sección de recursos, y en el aula, avanzar en la resolución de los problemas planteados.
- Utilizar en el laboratorio, el IDE de Netbeans para codificar los programas en Java, respetando los principios de una buena programación.
- Entregar los programas, los cuales estarán organizados en una tabla de doble entrada, que incluya: el código fuente a partir del método main(), la captura de pantalla del

Tipo de actividad:

Aula (X) Plataforma () Laboratorio (X)
Grupal (X) Individual (X) Equipo ()
Independiente ()

Recursos:

- Listado de ejercicios usando arreglos unidimensionales (vectores)
- Video: [Programación en Java. Arreglos. Definición de arreglos](#)
- Video: [Arreglos de tipo String en Java - Manejo de cadenas de arreglos en Java](#)
- Aplicación: Word o aplicación de acceso libre en Internet
- IDE: Netbeans
- Libro: [Java 2: iniciación y referencia](#). Sánchez Allende, J.

<p>código fuente completo y la salida a consola. Cada tabla tendrá como encabezado el enunciado del problema y será una para cada programa entregado.</p> <ul style="list-style-type: none"> • Integrar un documento en Word, o en una aplicación de acceso libre en Internet, con una portada, las tablas de cada programa y las fuentes consultadas en formato APA. • Convertir a PDF el documento y subirlo a plataforma. <p>4 hrs. Aula 1 hr. Laboratorio</p>	<p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Cumplir con los criterios que se indican en la tabla de doble entrada (etiquetas): código fuente (primera columna) y captura de pantalla del código fuente completo y salida a consola (segunda columna). • Entregar el total de los programas solicitados. • Entregar el total de los programas codificados: Sin errores de sintaxis. Sin errores lógicos. Con todos los requerimientos específicos de entrada y salida de datos. Sin errores ortográficos en el código fuente. • Sin errores ortográficos en el documento integrado.
---	--

EC3 Fase II: Arreglos bidimensionales (matrices).

Contenido: Declaración, inicialización, almacenamiento, recuperación y recorrido en arreglos bidimensionales o matrices.

<p>EC3 F2 Actividad de aprendizaje 17: Programas en Java usando arreglos bidimensionales (matrices).</p> <p>Desarrollar en binas programas en Java usando como estructura de almacenamiento un arreglo bidimensional o matriz, para dar solución a los problemas planteados que proporcione el facilitador.</p> <ul style="list-style-type: none"> • Revisar en la clase presencial los videos y el capítulo 6, que se aparecen en la sección de recursos y avanzar el la resolución de problemas planteados. • Utilizar en el laboratorio, el IDE de Netbeans para codificar los programas en Java, respetando los principios de una buena programación. • Entregar los programas, los cuales estarán organizados en una tabla de doble entrada que incluya: el código fuente a partir del método main(), la captura de pantalla del código fuente completo y la salida a consola. Cada tabla tendrá como encabezado el enunciado del problema y será una para cada programa entregado. • Integrar un documento en Word, o en una aplicación de acceso libre en Internet, con una portada, las tablas de cada programa y las fuentes consultadas en formato APA. • Cuidar la ortografía, tanto en la escritura de los programas, con en el documento en general. • Convertir a PDF el documento y subirlo a plataforma. <p>4 hrs. Aula 2 hrs. Plataforma</p>	<p>Tipo de actividad: Aula (X) Plataforma (X) Laboratorio () Grupal (X) Individual () Equipo (X) Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Listado de programas en Java usando arreglos bidimensionales (matrices) • Video: Curso Java - 5: Arreglos y matrices (Arrays) • Aplicación: Word o aplicación de acceso libre en Internet • IDE: Netbeans • Libro: Java 2: iniciación y referencia. Sánchez Allende, J. <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Cumplir con los criterios que se indican en la tabla de doble entrada: código fuente (primera columna) y captura de pantalla del código fuente completo y salida a consola (segunda columna). • Entregar el total de los programas solicitados. • Entregar el total de los programas codificados: Sin errores de sintaxis. Sin errores lógicos. Con todos los requerimientos específicos de entrada y salida de datos. Sin errores ortográficos en el código fuente. • Sin errores ortográficos en el documento integrado.
--	--

<p>EC3 F2 Actividad de aprendizaje 18: Evaluación integradora sobre arreglos en Java (vectores y matrices).</p> <p>Desarrollar una evaluación integradora sobre arreglos en Java.</p> <ul style="list-style-type: none"> • Analizar detalladamente el problema que plantee el facilitador en la evaluación y desarrollar un programa en Java en el IDE de Netbeans, que lo solucione. • Usar la estructura de almacenamiento tipo arreglo, así como, las estructuras de control secuenciales, selectivas y repetitivas que apliquen a la solución, respetando los principios de una buena programación. • Crear el proyecto y ponerle el nombre según se muestra en el ejemplo: nombreApellidoPaternoExamen, por ejemplo, "juanAlcantarExamen.java". • Respetar los principios de una buena programación. • Cumplir los criterios de evaluación. • Finalizar la evaluación en el tiempo establecido y enviar el código fuente a la plataforma educativa. <p>2 hrs. Laboratorio</p>	<p>Tipo de actividad: Aula () Plataforma () Laboratorio (X) Grupal () Individual (X) Equipo () Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Evaluación • IDE: Netbeans <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Entregar el programa de la evaluación codificado en Java: Sin errores de sintaxis. Sin errores lógicos. Con todos los requerimientos específicos de entrada y salida de datos. • Sin errores ortográficos en el código fuente. • Respetando el tiempo establecido de 2 hrs. • Y el criterio establecido para el nombre (juanAlcantarExamen.java).
<p>EC3 Fase III: Introducción a los métodos en Java.</p> <p>Contenido: Definición y uso de métodos, declaración y llamada de métodos de clase o estáticos tipo void con y sin parámetros.</p>	
<p>EC3 F3 Actividad de aprendizaje 19: Apuntes digitales sobre los métodos en Java.</p> <p>Desarrollar en equipo, apuntes digitales sobre los métodos en Java.</p> <ul style="list-style-type: none"> • Organizar la información usando diversos esquemas (tablas, cuadros sinópticos, mapas conceptuales, etc.), listas numeradas y/o imágenes. Para el diseño de los esquemas, apoyarse en las herramientas de ilustraciones de Word o en alguna aplicación de distribución libre disponible en Internet. • Incluir en los apuntes digitales la siguiente información: Métodos (definición, uso, tipos de métodos: de objeto y de clase). ¿Para qué se utilizan las palabras reservadas <i>static</i> y <i>void</i> en los métodos de clase? ¿Qué son los parámetros? Declaración de un método de clase tipo <i>void</i> con y sin parámetros (sintaxis y un ejemplo de cada uno). Llamada de un método de clase tipo <i>void</i> con o sin parámetros (un ejemplo de cada uno). • Revisar el capítulo 5 del libro recomendado en 	<p>Tipo de actividad: Aula (X) Plataforma (X) Laboratorio () Grupal () Individual () Equipo (X) Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Aplicación: Word o aplicación de distribución libre en Internet • Libro: Empezar a programar usando Java . Prieto, N. <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Cumplir con todos los temas solicitados. • Utilizar diversos esquemas, listas y/o imágenes para organizar la información. • Ordenar los apuntes para su comprensión. • Utilizar al menos una referencia, que corresponda a las fuentes de información sugeridas en la secuencia didáctica. • Escribir citas y referencias en formato APA y

<p>los recursos y apoyarse en otras fuentes confiables.</p> <ul style="list-style-type: none"> • Integrar un documento que incluya una portada, los apuntes digitales y las referencias en formato APA. • Convertir a PFD el documento y subirlo a la plataforma. • Concluir el tema, mediante sesión de preguntas y respuestas que realizará el profesor para aclarar y/o reforzar los temas en el salón de clases. <p>1 hr. Aula 1 hr. Plataforma</p>	<p>respetar las reglas ortográficas.</p>
<p>EC3 F3 Actividad de aprendizaje 20: Programas en Java usando métodos de clase o estáticos.</p> <p>Desarrollar en el laboratorio programas en Java usando métodos de clase o estáticos, con y sin parámetros, para dar solución a los problemas planteados, que proporcione el facilitador.</p> <ul style="list-style-type: none"> • Revisar en la clase, el video y el capítulo 5, que se aparecen en la sección de recursos. • Utilizar el IDE de Netbeans para codificar los programas en Java, respetando los principios de una buena programación. • Entregar los programas, los cuales estarán organizados en una tabla de doble entrada que incluya: el código fuente completo y la captura de pantalla del mismo incluyendo la salida a consola. Cada tabla tendrá como encabezado el enunciado del problema y será una para cada programa entregado. • Integrar un documento en Word o en una aplicación de distribución libre en Internet, con una portada, las tablas de cada programa y las fuentes consultadas en formato APA. • Convertir a PDF el documento y subirlo a plataforma. <p>3 hrs. Aula 1 hr. Laboratorio</p>	<p>Tipo de actividad: Aula (X) Plataforma () Laboratorio (X) Grupal (X) Individual (X) Equipo () Independiente ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Listado de programas en Java, usando métodos de clase o estáticos. • Video: 70. Programación en Java POO Miembros estáticos de una clase • Aplicación: Word o aplicación de distribución libre en Internet • IDE: Netbeans • Libro: Empezar a programar usando Java. Prieto, N. <p>Criterios de evaluación de la actividad:</p> <ul style="list-style-type: none"> • Cumplir con los criterios que se indican en la tabla de doble entrada: código fuente completo (primera columna) y captura de pantalla del código fuente completo y salida a consola (segunda columna). • Entregar el total de los programas solicitados. • Entregar el total de los programas codificados: Sin errores de sintaxis. Sin errores lógicos. Con todos los requerimientos específicos de entrada y salida de datos. Sin errores ortográficos en el código fuente. • Sin errores ortográficos en el documento integrado.
<p>Evaluación formativa:</p> <ol style="list-style-type: none"> 1. Cuadro de triple entrada sobre la estructura de almacenamiento arreglos en Java. 2. Ejercicios en Java usando arreglos unidimensionales (vectores). 3. Programas en Java usando arreglos bidimensionales (matrices). 4. Examen integrador sobre arreglos en Java (vectores y matrices). 5. Apuntes digitales sobre los métodos en Java. 	

6. Programas en Java usando métodos de clase o estáticos.

Fuentes de información

1. Byspel. (3 de julio de 2018). *Arreglos de tipo string en Java - Manejo de cadenas de arreglos en Java*. [Video]. YouTube. <https://youtu.be/JsF5O6QeefM>
2. Ceballos Sierra, F. J. (2015). *Java 2: lenguaje y aplicaciones*. RA-MA Editorial. <https://elibro.net/es/lc/ues/titulos/62458>
3. Códigos de Programación. (23 de marzo de 2017). *Curso Java - 5: Arreglos y Matrices (Arrays)*. [Video]. YouTube. <https://youtu.be/KZwMRL2q6O8>
4. Deitel, P. (2016). *Cómo programar en Java*. Pearson.
8. Programación ATS. (2 de abril de 2017). *Programación en Java || Arreglos || Definición de Arreglos*. [Video]. YouTube. <https://youtu.be/AZc3hxMxl54?t=11>
9. Programación ATS. (19 de julio de 2017). *70. Programación en Java || POO || Miembros estáticos de una clase*. [Video]. YouTube. <https://youtu.be/aiBzXQmtzrA>
10. Sánchez Allende, J. (2005). *Java 2: iniciación y referencia*. McGraw-Hill. <https://elibro.net/es/ereader/ues/50290?page=156>
11. Sznajdleder, P. (2020). *Java a fondo*. Alfaomega.

Políticas

Generales

- Los teléfonos celulares deberán permanecer sin sonido en la clase y apagados en los exámenes.
- Las participaciones en clase serán para enriquecer la misma y/o para aclarar dudas.
- Se deberá mostrar respeto dentro del aula, a los compañeros y el profesor, incluso a través del uso de medios digitales de comunicación.
- Se tendrá un comportamiento ético, en relación al desarrollo de las actividades y comportamiento en general.

Asistencia

- Se tomará lista diariamente.
- Se permitirá una tolerancia de 10 minutos para entrar a

Metodología

- Es responsabilidad del estudiante gestionar los procedimientos necesarios para alcanzar el desarrollo de las competencias del curso.
- El curso se desarrollará combinando sesiones presenciales y virtuales, así como prácticas presenciales en laboratorios, campos o a distancia en congruencia con la naturaleza de la asignatura.
- Los productos académicos escritos deberán ser entregados en formato PDF en la plataforma institucional.
- Todas las actividades, tienen una fecha de inicio y una de fin, las cuales deberán ser respetadas para avanzar apropiadamente con los tiempos definidos en el semestre, en pro del logro de la competencia del curso.

Evaluación

Artículo 27. La evaluación es el proceso que permite valorar el desarrollo de las competencias establecidas en las secuencias didácticas del plan de estudio del programa educativo correspondiente. Su metodología es integral y considera diversos tipos de evidencias de conocimiento, desempeño y producto por parte del alumno.

Artículo 28. Las modalidades de evaluación en la Universidad son:

I. Diagnóstica permanente, entendiéndola esta como la evaluación continua del estudiante durante la realización de una o varias actividades;

II. Formativa, siendo esta, la evaluación al alumno durante el desarrollo de cada elemento de competencia; y

III. Sumativa es la evaluación general de todas y cada una de las actividades y evidencias de las secuencias didácticas. Sólo los resultados de la evaluación

clases.

- Las inasistencias serán justificadas, mediante comprobante oficial o a criterio del profesor, siempre y cuando el estudiante se haya comunicado oportunamente y se establezcan las estrategias de recuperación de los temas tratados y de las actividades asignadas o desarrolladas.

Será responsabilidad del estudiante

- Revisar la plataforma educativa con regularidad.
- Entregar las actividades en la plataforma educativa en la hora y fecha establecidas.
- Cumplir con todos los criterios de evaluación de la actividad.
- Cumplir con todas las actividades establecidas en la evaluación formativa.

sumativa tienen efectos de acreditación y serán reportados al departamento de registro y control escolar.

Artículo 29. La evaluación sumativa será realizada tomando en consideración de manera conjunta y razonada, las evidencias del desarrollo de las competencias y los aspectos relacionados con las actitudes y valores logrados por el alumno. Para tener derecho a la evaluación sumativa de las asignaturas, el alumno deberá:

I. Cumplir con la evidencia de las actividades establecidas en las secuencias didácticas;

II. Asistir como mínimo al 70% de las sesiones de clase impartidas.

Artículo 30. Los resultados de la evaluación expresarán el grado de dominio de las competencias, por lo que la escala de evaluación contemplará los niveles de:

- I. Competente sobresaliente;
- II. Competente avanzado;
- III. Competente intermedio;
- IV. Competente básico; y
- V. No aprobado.

El nivel mínimo para acreditar una asignatura será el de competente básico. Para fines de acreditación los niveles tendrán un equivalente numérico conforme a la siguiente tabla:

Artículo 31. Para lograr la acreditación de las competencias comprendidas en las secuencias didácticas de las asignaturas del programa educativo, el alumno dispondrá de los siguientes medios:

Competente sobresaliente	10
Competente avanzado	9
Competente intermedio	8
Competente básico	7
No aprobado	6

I. La evaluación sumativa, mínimo

7, competente básico;
II. La demostración de competencias previamente adquiridas;
III. Por convalidación, revalidación o equivalencia.

Artículo 32. Los resultados de la evaluación sumativa serán dados a conocer a los alumnos, en un plazo no mayor de cinco días hábiles después de concluido el proceso.

Artículo 33. En caso de que el alumno considere que existe error u omisión en el registro de evaluación sumativa, podrá presentar solicitud por escrito ante el director de la unidad académica dentro de los cinco días hábiles siguientes contados a partir de la fecha de publicación de los resultados, quien en igual termino emitirá una respuesta.