



Guía para el Profesor

COMPUTACIÓN EN JAVA

LSTI2001

Profesional

Índice

Información general del curso	3
Competencia del curso	3
Metodología.....	3
Evaluación	4
Bibliografía y recursos especiales	6
Contenido del curso (temas).....	6
Uso de rúbricas	12
Tips importantes	13

Información general del curso

- Profesional
- Plan académico 2013 y 2018.
- Modalidades
 - Clave banner: LSTI2001.
 - Modalidad: Presencial y Online.
 - Frecuencia de clases: 3 horas por semana.
 - Técnica didáctica: Aprendizaje basado en proyecto.

Competencia del curso

Utiliza la tecnología y el lenguaje Java® para instalar y configurar un ambiente de desarrollo eficiente mediante la programación estructurada y orientada a objetos, integrando diferentes tecnologías que le sean relevantes.

Metodología

El curso está integrado por 12 temas. En cada tema, se encontrará:

- Una breve explicación del tema que ayudará a ampliar el conocimiento.
- Una serie de lecturas y videos que se deben revisar de manera obligatoria para una mejor comprensión de los temas.
- Una lista de lecturas y videos que se recomiendan para complementar el estudio del tema.
- Una actividad de aprendizaje cuyo propósito es aplicar y experimentar con los conceptos estudiados.

A lo largo del curso se debe trabajar en lo siguiente:

- 13 Actividades
- 1 Evidencia

Actividades

Las actividades se deben enviar a través de la plataforma Canvas en la fecha marcada en la agenda publicada desde el primer día del curso. Es importante que se respeten estas fechas, por lo que se recomienda que se elabore un plan personal que ayude a programar las tareas.

Evidencia

La evidencia se subdivide en tres entregables, que consiste en crear un cliente de Facebook escrito en Java® con APIs Open Source como Facebook4J y Log4j2 aplicando los conocimientos adquiridos en el curso, desde el diseño (diagrama de flujo y pseudocódigo) con base en los requerimientos preestablecidos, investigación de desarrollo de aplicaciones en Facebook y permisos del API, hasta el manejo de versiones de código en GitHub. Si no se tiene cuenta de Facebook se recomienda una con propósitos de prueba, sin embargo, se puede realizar algo similar con Twitter con el API Twitter4j, siguiendo los lineamientos de la evidencia. La Evidencia se subdivide en tres entregables.

Importante: se debe revisar con tiempo los criterios de evaluación para las dos fases iniciales que son la entrega 1: Ambiente de desarrollo y requerimientos; de igual forma para el avance 2: Investigación, diseño y programación. Ambos vienen en las instrucciones de la evidencia.

Para la entrega final: Programación y documentación. Se deberá revisar la rúbrica con la cual se evaluará.

GUÍA PARA EL PROFESOR

Evaluación

Unidades	Instrumento evaluador	Puntos
13	Actividades	60
1	Evidencia	40
Total		100

Tema	Actividad	Ponderación
Tema 1	Actividad 1	2
Tema 2	Actividad 2	2
Tema 3	Actividad 3	2
Tema 4	Actividad 4	2
Tema 5	Actividad 5.	2
Avance de evidencia 1		10
Tema 6	Actividad 6	4
Tema 7	Actividad 7	4
Tema 8	Actividad 8	6
Tema 9	Actividad 9	6
Avance de evidencia 2		10
Tema 10	Actividad 10	7
Tema 11	Actividad 11	7
Tema 12	Actividad 12	8
Tema 13	Actividad 13.	8
Entrega final		20
		100 puntos

Bibliografía y recursos especiales

Libros de texto

- Downey, A. B., Mayfield, C. (2019). *Think Java, 2nd Edition*. Recuperado de <https://books.trinket.io/thinkjava2/>

Libros de apoyo

- Evans, B., Evans, B. J., & Flanagan, D. (2019). *Java in a Nutshell (7ª ed.)*. United States: O'Reilly Media, Incorporated. ISBN 1492037257

Requisitos especiales

Requisitos especiales	Especificación	Temas en los que se usará
Sala de cómputo (Mac o PC)	PC	Tema 1-13
Software	IntelliJ IDEA Java JDK 11 Git SCM	Tema 1-13

Contenido del curso (temas)

Módulo 1.

Tema 1. Introducción a la programación.

Objetivo:

Comprender la programación orientada a objetos, el ciclo de vida del software desde la detección de errores, mantenimiento y principios de la programación estructurada.

Notas para la enseñanza del tema:

Leer el primer capítulo del libro de la clase:

- Downey, A. B., & Mayfield, C. (2019). Chapter 1 Computer Programming. *Think Java: How to Think Like a Computer Scientist (2ª ed.)* (pp.1-13). California, United States: O'Reilly Media, Incorporated.

Ver los videos incluidos en el área de cinema

Revisar algunos ejercicios de solución de problemas básicos con diagramas de flujo y pseudocódigo como: números primos, pares, impares, etc.

Notas para la actividad:

Esta actividad no requiere codificar, ni condiciones detalladas en el algoritmo, diagrama de flujo o pseudocódigo.

Proporcionar varios ejemplos de palíndromos.

Ejemplo de un pseudocódigo:

- Pedir una palabra.
- Iniciar una variable temporal para almacenar la palabra escrita a la inversa.
- Procesar cada una de las letras para almacenarla en la variable temporal a la inversa.
 - Iniciar un ciclo que convierta la palabra a la inversa.
 - Ciclo inicia en la longitud menos uno

GUÍA PARA EL PROFESOR

- Ciclo termina en cero.
- Ciclo disminuye
- La variable temporal a la inversa se equivale a si misma más la palabra de la original.
- Comparar la palabra original con la palabra a la inversa letra por letra.
- Si es palíndromo, informar al usuario.

Tema 2. La tecnología Java.

Objetivo:

Definir la tecnología Java®, presencia, plataformas y practicar con la instalación y configuración del Java® Development Kit.

Notas para la enseñanza del tema:

El maestro/tutor deberá estar familiarizado con los diferentes ambientes donde Java puede ser ejecutado: Windows, Linux, Mac, Solaris.

Se recomienda estar familiarizado con variables de ambiente en los diferentes sistemas operativos.

Definición de variables de ambiente de Java: JAVA_HOME, CLASSPATH.

Leer la documentación de Java SE 11 (LTS) en <https://www.oracle.com/java/technologies/javase-downloads.html#JDK11>

Notas para la actividad:

Revisar el proceso de instalación y configuración de Java 11 en <https://docs.oracle.com/en/java/javase/11/install/overview-jdk-installation.html#GUID-8677A77F-231A-40F7-98B9-1FD0B48C346A>

Entender el funcionamiento de los comandos `java`, `javac` y `javap`

Tema 3. Ambientes de desarrollo 1.

Objetivo:

Después de conocer las ventajas de los ambientes de desarrollo integrados, seleccionar el de su preferencia basado en el criterio del alumno y tomando en cuenta que el que será usado en este curso es IntelliJ IDEA.

Notas para la enseñanza del tema:

El reporte de Rebellabs puede mostrar una perspectiva del impacto de los IDE tanto por el uso como por la integración con demás tecnologías en <https://www.jrebel.com/blog/2020-java-technology-report>

Revisar la instalación y configuración de IntelliJ IDEA

Notas para la actividad:

Es recomendable que durante la impartición del tema, se sigan los pasos de instalación y configuración de IntelliJ IDEA para aclarar posibles preguntas del participante.

Tema 4. Ambientes de desarrollo 2.

Objetivo:

Después de configurar adecuadamente el ambiente de desarrollo y el entorno de desarrollo integrado, el estudiante instalará y configurará el sistema de control de versiones Git.

Notas para la enseñanza del tema:

Se recomienda realizar la lectura previa de las referencias bibliográficas recomendadas.

Ver el video de Linus Torvalds (<https://www.youtube.com/watch?v=o8NPlizkFhE>), donde se explica la motivación detrás de Git.

El siguiente video:

- Google (2007, 14 Mayo) Youtube. Tech Talk: Linus Torvalds on git [Archivo de Video] Recuperado de <https://www.youtube.com/watch?v=4XpnKHJAok8>

GUÍA PARA EL PROFESOR

presenta una plática técnica y práctica sobre Git de la propia voz de Linus Torvalds; se trata de una conferencia realizada a desarrolladores de Google, el video dura 1 hora, es muy extenso para los alumnos, pero el docente puede obtener conocimiento valioso para compartir y comentar con los alumnos.

Separar media parte del tiempo dedicado el tema a instalar y configurar Git, de ser posible menos.

Dedicar la siguiente parte del tema a realizar los ejercicios del programa Git It, especificados en la práctica 1. Revisar el programa en <https://github.com/jlord/git-it-electron#git-it-desktop-app>

Notas para la actividad:

Es recomendable que durante la impartición del tema, se sigan los pasos de instalación y configuración de Git para aclarar posibles preguntas del participante.

Los alumnos deberán demostrar que han creado su cuenta de Github, Bitbucket o cualquier otro servicio de repositorios en línea.

Los alumnos deberán mostrar evidencia de que realizaron los ejercicios del programa Git It en su cuenta de Github/Bitbucket.

Tema 5. Lenguaje Java 1.

Objetivo:

Comprender la estructura del lenguaje Java®, conocer los tipos de datos primitivos, operadores, arreglos y strings.

Notas para la enseñanza del tema:

Analizar los videos incluidos en la sección de cinema.

Leer las lecturas de la bibliografía del tema y la documentación de Java:

- Downey, A. B., & Mayfield, C. (2019). Chapter 2 Variables and Operators. Think Java: How to Think Like a Computer Scientist (2ª ed.) (pp.17-33). California, United States: O'Reilly Media, Incorporated.
- Downey, A. B., & Mayfield, C. (2019). Chapter 7 Arrays and References. Think Java: How to Think Like a Computer Scientist (2ª ed.) (pp.107-126). California, United States: O'Reilly Media, Incorporated.
- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 2. Java Syntax from the Ground Up. Java in a Nutshell (7ª ed.) (pp.21-57). California, United States: O'Reilly Media, Incorporated.
- Oracle. (s.f.). How to Write Doc Comments for the Javadoc Tool. Recuperado 14 marzo, 2020, de <http://www.oracle.com/technetwork/articles/java/index-137868.html>
- Oracle. (s.f.). Assignment, Arithmetic, and Unary Operators. Oracle. Recuperado 14 marzo, 2020, de <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op1.html>
- Oracle. (s.f.). Equality, Relational, and Conditional Operators. Oracle. Recuperado 14 marzo, 2020, de <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op2.html>
- Oracle. (s.f.). Bitwise and Bit Shift Operators. Oracle. Recuperado 14 marzo, 2020, de <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op3.html>
- Oracle. (s.f.). Java Language Keywords. Recuperado 14 marzo, 2020, de <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/keywords.html>
- Oracle. (s.f.). Primitive Data Types. Oracle. Recuperado 14 marzo, 2020, de <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
- Oracle. (s.f.). Operators. Oracle. Recuperado 14 marzo, 2020, de <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html>
- Oracle. (s.f.). Summary of Operators. Oracle. Recuperado 14 marzo, 2020, de <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/opsummary.html>

Explorar los ejemplos mostrados durante el módulo.

Lo ideal sería ejecutar cada uno de los ejemplos durante la explicación.

Motivar a los alumnos a modificar los ejemplos del tema para experimentar.

Notas para la actividad:

Los alumnos deberán crear un nuevo repositorio para la actividad y subir el código correspondiente.

Módulo 2.

Tema 6. Programación orientada a objetos.

Objetivo:

Definir el concepto de clases, objetos, variables, métodos y operadores de acceso.

Notas para la enseñanza del tema:

Analizar los videos incluidos en la sección de cinema.

Leer las lecturas de la bibliografía del tema y la documentación de Java:

- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 3. Object-Oriented Programming in Java. Java in a Nutshell (7ª ed.) (pp.120-166). California, United States: O'Reilly Media, Incorporated.
- Downey, A. B., & Mayfield, C. (2019). Chapter 11. Designing Classes. Think Java: How to Think Like a Computer Scientist (2ª ed.) (pp.183-197). California, United States: O'Reilly Media, Incorporated.

Explorar los ejemplos mostrados durante el módulo.

Lo ideal sería ejecutar cada uno de los ejemplos durante la explicación.

Motivar a los alumnos a modificar los ejemplos para experimentar.

Notas para la actividad:

Los alumnos deberán crear un nuevo repositorio para la actividad y subir el código correspondiente.

Tema 7. Instrucciones de control de flujo.

Objetivo:

Definir los tipos de instrucciones de control de flujo y practicar constantemente con código.

Notas para la enseñanza del tema:

Analizar los videos incluidos en la sección de cinema.

Leer las lecturas de la bibliografía del tema y la documentación de Java:

- Downey, A. B., & Mayfield, C. (2019). Chapter 5. Conditionals and Logic. Think Java: How to Think Like a Computer Scientist (2ª ed.) (pp.71-88). California, United States: O'Reilly Media, Incorporated.
- Downey, A. B., & Mayfield, C. (2019). Chapter 6. Loops and Strings. Think Java: How to Think Like a Computer Scientist (2ª ed.) (pp.89-106). California, United States: O'Reilly Media, Incorporated.
- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 2. Java Syntax from the Ground Up. Java in a Nutshell (7ª ed.) (pp.65-76). California, United States: O'Reilly Media, Incorporated.
- Oracle. (s.f.-b). The if-then and if-then-else Statements The Java™ Tutorials. Recuperado 23 marzo, 2020, de <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>
- Oracle. (s.f.-b). The switch Statement The Java™ Tutorials. Recuperado 23 marzo, 2020, de <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>
- Oracle. (s.f.-c). The while and do-while Statements The Java™ Tutorials. Recuperado 26 marzo, 2020, de <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html>
- Oracle. (s.f.-d). The for Statement The Java™ Tutorials. Recuperado 26 marzo, 2020, de <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html>
- Oracle. (s.f.-e). Branching Statements The Java™ Tutorials. Recuperado 26 marzo, 2020, de <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/branch.html>

Explorar los ejemplos mostrados durante el módulo.

Lo ideal sería ejecutar cada uno de los ejemplos durante la explicación.

Motivar a los alumnos a modificar los ejemplos para experimentar.

Notas para la actividad:

Los alumnos deberán crear un nuevo repositorio para la actividad y subir el código correspondiente.

Tema 8. Lenguaje Java 2.

Objetivo:

Que el alumno comprenda la estructura de los paquetes principales del API de Java®, conocer algunas clases internas y aprender a utilizar los paquetes para facilitar el desarrollo.

Notas para la enseñanza del tema:

Analizar los videos incluidos en la sección de cinema.

Leer las lecturas de la bibliografía del tema y la documentación de Java:

- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 2. Java Syntax from the Ground Up. Java in a Nutshell (7ª ed.)(pp.109-110). California, United States: O'Reilly Media, Incorporated.
- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 8. Working with Java Collections. Java in a Nutshell (7ª ed.)(pp.303-336). California, United States: O'Reilly Media, Incorporated.
- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 9. Handling Common Data Formats. Java in a Nutshell (7ª ed.)(pp.337-360). California, United States: O'Reilly Media, Incorporated.
- Oracle. (s. f.-c). Package java.lang. Recuperado 5 de abril de 2020, de <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/package-summary.html>
- Oracle. (s. f.-c). Package java.util. Recuperado 5 de abril de 2020, de <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/package-summary.html>

Explorar los ejemplos mostrados durante el módulo.

Lo ideal sería ejecutar cada uno de los ejemplos durante la explicación.

Motivar a los alumnos a modificar los ejemplos para experimentar.

Notas para la actividad:

Los alumnos deberán crear un nuevo repositorio para la actividad y subir el código correspondiente.

Tema 9. Entrada y salida de datos 1.

Objetivo:

Que el alumno comprenda las diferentes formas que existen para proporcionar datos a un programa desde la línea de comandos o pidiendo los datos al usuario desde la entrada estándar

Notas para la enseñanza del tema:

Entender el funcionamiento de la línea de comandos en los diferentes sistemas operativos: Windows, OSX y Linux.

Analizar los videos incluidos en la sección de cinema.

Leer las lecturas de la bibliografía del tema y la documentación de Java:

- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 10. File Handling and I/O. Java in a Nutshell (7ª ed.)(pp.361-384). California, United States: O'Reilly Media, Incorporated.
- Downey, A. B., & Mayfield, C. (2019). Chapter 3 Input and Output. Think Java: How to Think Like a Computer Scientist (2ª ed.) (pp.33-50). California, United States: O'Reilly Media, Incorporated.
- JetBrains. (s. f.). Add environment variables and program arguments - Help | AppCode. Recuperado 20 de abril de 2020, de <https://www.jetbrains.com/help/objc/add-environment-variables-and-program-arguments.html>
- Oracle. (s. f.-b). Command-Line Arguments. Recuperado 20 de abril de 2020, de <https://docs.oracle.com/javase/tutorial/essential/environment/cmdLineArgs.html>

Explorar los ejemplos mostrados durante el módulo.

Lo ideal sería ejecutar cada uno de los ejemplos durante la explicación.

Motivar a los alumnos a modificar los ejemplos para experimentar.

Notas para la actividad:

Los alumnos deberán crear un nuevo repositorio para la actividad y subir el código correspondiente.

Módulo 3.

Tema 10. Manejo de excepciones.

Objetivo:

Comprender las excepciones en tiempo de ejecución en una aplicación de Java® así como los efectos de excepciones sin atrapar y con esto aplicar las sentencias try-catch para interpretar los mensajes de error.

Notas para la enseñanza del tema:

Analizar los videos incluidos en la sección de cinema.

Leer las lecturas de la bibliografía del tema y la documentación de Java:

- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 2. Java Syntax from the Ground Up. Java in a Nutshell (7ª ed.)(pp.78-84). California, United States: O'Reilly Media, Incorporated.
- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 2. Java Syntax from the Ground Up. Java in a Nutshell (7ª ed.)(pp.89-90). California, United States: O'Reilly Media, Incorporated.
- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 5. Introduction to Object-Oriented Design in Java. Java in a Nutshell (7ª ed.)(pp.245-246). California, United States: O'Reilly Media, Incorporated.
- Downey, A. B., & Mayfield, C. (2019). Chapter 15. Arrays of Arrays. Think Java: How to Think Like a Computer Scientist (2ª ed.) (pp.260-261). California, United States: O'Reilly Media, Incorporated.

Explorar los ejemplos mostrados durante el módulo.

Lo ideal sería ejecutar cada uno de los ejemplos durante la explicación.

Motivar a los alumnos a modificar los ejemplos para experimentar.

Notas para la actividad:

Los alumnos deberán crear un nuevo repositorio para la actividad y subir el código correspondiente.

Tema 11. Conceptos avanzados de POO.

Objetivo:

Comprender los conceptos avanzados de la programación orientada a objetos en Java® para aplicarlos en el diseño de programas y con esto mejorar la organización, escalabilidad y capacidad de mantenimiento.

Notas para la enseñanza del tema:

Analizar los videos incluidos en la sección de cinema.

Leer las lecturas de la bibliografía del tema y la documentación de Java:

- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 2. Java Syntax from the Ground Up. Java in a Nutshell (7ª ed.) (p.86). California, United States: O'Reilly Media, Incorporated.
- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 2. Java Syntax from the Ground Up. Java in a Nutshell (7ª ed.) (pp.110-119). California, United States: O'Reilly Media, Incorporated.
- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 3. Object-Oriented Programming in Java. Java in a Nutshell (7ª ed.) (pp.130-133). California, United States: O'Reilly Media, Incorporated.
- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 3. Object-Oriented Programming in Java. Java in a Nutshell (7ª ed.) (pp.144-163). California, United States: O'Reilly Media, Incorporated.
- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 3. Object-Oriented Programming in Java. Java in a Nutshell (7ª ed.) (pp.168-177). California, United States: O'Reilly Media, Incorporated.
- Downey, A. B., & Mayfield, C. (2019). Chapter 14: Extending Classes. Think Java: How to Think Like a Computer Scientist (2ª ed.) (pp.239-240). California, United States: O'Reilly Media, Incorporated.
- Downey, A. B., & Mayfield, C. (2019). Chapter 16: Reusing Classes. Think Java: How to Think Like a Computer Scientist (2ª ed.) (pp.273-275). California, United States: O'Reilly Media, Incorporated.
- Downey, A. B., & Mayfield, C. (2019). Chapter 17: Advanced Topics. Think Java: How to Think Like a Computer Scientist (2ª ed.) (pp.290-291). California, United States: O'Reilly Media, Incorporated.

Explorar los ejemplos mostrados durante el módulo.
Lo ideal sería ejecutar cada uno de los ejemplos durante la explicación.
Motivar a los alumnos a modificar los ejemplos para experimentar.

Notas para la actividad:

Los alumnos deberán crear un nuevo repositorio para la actividad y subir el código correspondiente.

Tema 12. Entrada y salida de datos 2.

Objetivo:

Comprender los conceptos generales del manejo de archivos, la lectura y escritura de datos, y la manipulación de archivos y carpetas.

Notas para la enseñanza del tema:

Analizar los videos incluidos en la sección de cinema.

Leer las lecturas de la bibliografía del tema y la documentación de Java:

- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 10. File Handling and I/O. Java in a Nutshell (7ª ed.) (pp.361-384). California, United States: O'Reilly Media, Incorporated.

Explorar los ejemplos mostrados durante el módulo.
Lo ideal sería ejecutar cada uno de los ejemplos durante la explicación.
Motivar a los alumnos a modificar los ejemplos para experimentar.

Notas para la actividad:

Los alumnos deberán crear un nuevo repositorio para la actividad y subir el código correspondiente.

Tema 13. Expresiones lambda.

Objetivo:

Introducir las expresiones lambda y los métodos de referencia en Java®.

Notas para la enseñanza del tema:

Analizar los videos incluidos en la sección de cinema.

Leer las lecturas de la bibliografía del tema y la documentación de Java:

- Evans, B., Evans, B. J., & Flanagan, D. (2019). Chapter 4. The Java Type System. Java in a Nutshell (7ª ed.)(pp.198-223). California, United States: O'Reilly Media, Incorporated.

Explorar los ejemplos mostrados durante el módulo.
Lo ideal sería ejecutar cada uno de los ejemplos durante la explicación.
Motivar a los alumnos a modificar los ejemplos para experimentar.

Notas para la actividad:

Los alumnos deberán crear un nuevo repositorio para la actividad y subir el código correspondiente.

Uso de rúbricas

La evidencia final del curso tiene asignada una rúbrica con la cual es obligatorio que se califique, esto es muy importante para nuestro modelo de competencias ya que es la forma en la que medimos el desarrollo de las competencias en nuestros alumnos.

Es importante evaluar con la rúbrica que aparece en el apartado de la evidencia final ya que se les estará auditando constantemente su realización efectiva.

Video disponible para calificar con rúbricas en:

GUÍA PARA EL PROFESOR

- ¿Cómo busco una rúbrica?: <https://youtu.be/QgDKeZv9tAI>
- ¿Cómo califico con una rúbrica?: <https://youtu.be/mAbIsLAGlp4>

Tips importantes

- **Material de capacitación en la plataforma tecnológica Canvas:**
 - Tutorial digital para profesores: <https://bit.ly/2SbMaNK>
 - Tutorial digital para alumnos: <https://bit.ly/35IBnPg>
- **¿En dónde o a quién reporto un error detectado en el contenido del curso?**

Lo puedes reportar a la cuenta atencioncursos@servicios.tecmilenio.mx pero te pedimos que también reportes sugerencias para el contenido y actividades del curso.

- **¿Quién me informa de la cantidad de sesiones y tiempo de cada sesión en las semanas?**

El coordinador docente te debe de proporcionar esta información.

- **¿En qué semanas se aplican los exámenes parciales y el examen final?**

Consulta con tu coordinador docente los calendarios de acuerdo a la modalidad de impartición.

- **¿Tengo que capturar las calificaciones en Banner y en la plataforma educativa?**

Si, es importante que captures calificaciones en la plataforma para que los alumnos estén informados de su avance y reciban retroalimentación de parte tuya de todo lo que realizan en el curso. En banner es el registro oficial de las calificaciones de los alumnos.