MODALIDAD ABIERTA PLAN DE TRABAJO

DATOS DE LA ASIGNATURA

Licenciaturas en que se imparte:	Lic. Informática 6to semestre
Nombre:	Programación de dispositivos móviles
Clave(s):	1668
Tipo:	Obligatoria
Plan de Estudios:	**2012 (actualizado al 2016)**

FECHAS DEL SEMESTRE

Inicio semestre:	4 de febrero de 2025
Fin del semestre:	13 de junio 2025
Plataforma educativa:	19 de febrero de 2025 Primer día para entrega de actividades en plataforma
Cierre de plataformas:	25 de mayo de 2025 a las 23:00 hrs. Último día para entrega de actividades en plataforma
Periodo examen global:	6, 7, y del 9 al 12 de junio del 2025
Consulta de calificaciones en historia académica:	A partir del 30 de junio 2025

OBJETIVO GENERAL

CONTENIDO TEMÁTICO

Unidad	Tema	Teóricas
1	Introducción a las tecnologías móviles	1
2	Control y versionamiento con Git, Github y Git flow	1
3	Aprender sobre variables constantes, cadenas y operadores	1
4	Activities	25
5	Control de flujo, funciones y lambdas	12
6	Uso de colecciones en Java con Android	25
	Total de horas	**65**

BIENVENIDA

El alumno desarrollará aplicaciones móviles para Android por medio del lenguaje de programación Java.

Objetivos específicos:

El alumno:

1. Operará las sentencias de lenguaje de programación *Java* en proyectos para experimentar con código.
2. Configurará y manejará las secciones que componen el entorno de desarrollo integrado Android *Studio*.
3. Creará vistas con controles e implementará gestos con estos.
4. Implementará comunicación entre vistas para el paso de datos.
5. Comprender la aplicación de las interfaces en las aplicaciones.

PRESENTACIÓN DE LA ASIGNATURA

El curso está diseñado para que el alumno aprenda de manera práctica y teórica, interactuará con el lenguaje *Java* de manera amplia y conocerá las bases del lenguaje de programación con ejemplos reales y sus usos más comunes en el ámbito general del desarrollo de aplicaciones móviles.

El curso se dividirá en seis unidades de aprendizaje; teóricos y prácticos. En cada unidad habrá una o más actividades para el aprendizaje del estudiante.

Toda actividad deberá ser entregada en el Moodle académico proporcionada por la institución SUAYED.
https://github.com utilizando su propio repositorio y tendrá que invitar al profesor utilizando el correo electrónico cris52@comunidad.unam.mx.

Vea: *sugerencia de calendario de unidades de aprendizaje - S2025-2*

Vea: *sugerencia de calendario de exámenes - S2025-2*

Lineamientos importantes: actividades y exámenes

1. No se aceptará ninguna actividad fuera de tiempo sin algún motivo.
2. No hay reposición de exámenes.
3. Todas las actividades que lo ameriten deberán de llevar carátula, objetivo de la actividad, nombre completo, desarrollo y conclusiones, el trabajo es altamente académico y profesional de un futuro informático.

IMPORTANTE

4. **EN ESTA ASIGNATURA NO SE DEBERÁ UTILIZAR NINGUN CÓDIGO de programación de JETPACK COMPOSE.**

El alumno podrá comunicarse con el profesor en cualquiera de los siguientes medios para la aclaración de dudas o comentarios relacionados a la asignatura y/o en caso de alguna eventualidad externa.

- Slack©
  - Invitación: https://join.slack.com/t/fca-unam-info/shared_invite/zt-2k23kfsq9-yBsKeCVVKGtcZuODmcTO4g

FORMA EN QUE EL ALUMNADO DEBE PREPARAR LA ASIGNATURA

SUGERENCIA DE CALENDARIO DE ACTIVIDADES - 2025-2	
Fecha de entrega	Actividad
7 febrero 2025	Invitación https://github.com Valor, 4 pto
10-14 de febrero del 2025	M1-01 Valor, 4 pto
24-28 de febrero del 2025	M1-02 Valor, 4 pto
14 de marzo 2025	M1-03 Valor, 5 pto
4 de abril 2025	M2-01 Valor, 5 pto
14-18 de abril 2025	M3-01 Valor, 5 pto
25 de abril 2025	M4-01 Valor, 5 pto
2 de mayo 2025	M4-02 Valor, 5 pto
9 de mayo 2025	M5-01 Valor, 5 pto
12 de mayo 2025	M5-02

	Valor, 4 pto
19-23 de mayo 2025	M6-01 Valor, 4 pto

ACTIVIDADES POR REALIZAR DURANTE EL SEMESTRE

IV. SUGERENCIA DE CALENDARIO DE ACTIVIDADES ACTIVIDAD IMPORTANTE AL INICIAR EL CURSO - S2025-2	
Fecha de entrega	Actividad
Actividad importante y sugerida para actividades consecuentes.	Crear una cuenta en https://github.com utilizando su propio repositorio y tendrá que enviar una invitación al profesor utilizando el correo electrónico cris52@comunidad.unam.mx El medio de entregas oficiales es únicamente por la plataforma oficial de la Modalidad Abierta, en la cual deberá de agregar el enlace a https://github.com No se aceptan entregas por otro medio. Algunas actividades son seriadas, se sugiere realizarlas en tiempo y forma, de forma contraria no podrá entregar las actividades próximas.

Tabla de **Actividad-M1-01**		
Actividad	Instalación de Android Studio	
Objetivo del aprendizaje	Identificar el IDE y conocer las ventanas esenciales para el desarrollo en Android.	
Conocimientos previos requeridos	Sistemas operativos, lenguajes de programación.	
Tiempo	30 minutos	
Actividad sugerida:	1) Instalación de Android Studio en sistema operativo GNU/Linux. 2) Instalación de Android Studio en Windows. 3) Identifica las ventanas, haz un desglose y realiza captura de pantallas del visor de los resources, layout, assets, mipmap, logcat. 4) Publica tus resultados incluyendo las capturas de pantalla con su descripción. Incluir portada, nombre, actividad, fecha y una conclusión. Subir un sólo archivo en formato PDF en el repositorio de GitHub creado por el alumno. android_2025_m1_01.pdf	

Criterios de evaluación		

Criterios	Puntos máximos
Instalación en GNU/Linux	4
Instalación en Windows	4
Presentación y contenido	2
Total	10

Tabla de **Actividad-M1-02**		
Actividad	Hola Mundo	
Objetivo del aprendizaje	Ejecutar una sentencia básica de programación en lenguaje Java	
Conocimientos prev requeridos	Sistemas operativos, lenguajes de programación, instalación de Android Studio	
Actividad sugerida previa	**M1-01**	

<table>
<tr><td>Tiempo</td><td>15 minutos</td><td></td></tr>
<tr><td>Actividad sugerida:</td><td>1. Imprimir un Hello world.
2. Ingresar el código de programación. para imprimir en el log “Hello World”
3. Buscar la impresión del “hello world" en la ventana “logcat” dentro de Android Studio.
4. Incluir una captura de pantalla en el entregable.
5. Incluir portada, nombre, actividad, fecha, evidencia (capturas de pantalla) y una conclusión.

Subir un sólo archivo en formato PDF en el repositorio de GitHub creado por el alumno.

Subir actividad en el repositorio de GitHub creado por el alumno.
1. android_2025_m1_02.pdf
2. código fuente.</td><td></td></tr>
<tr><td>Criterios de evaluación</td><td colspan="2"><table>
<tr><td>Criterios</td><td>Puntos máximos</td></tr>
<tr><td>Generación de programa para que imprima su primer “Hola Mundo” en el logcat dentro de Android Studio.</td><td>4</td></tr>
</table></td></tr>
</table>

Despliegue de “Hola Mundo” en pantalla	5
Presentación y contenido	1
Total	10

Tabla de **Actividad-M1-03**		
Actividad	Arreglos y tiempo de ejecución	
Objetivo del aprendizaje	Ejecutar una sentencia básica de programación en lenguaje Java	
Conocimientos prev requeridos	Sistemas operativos, lenguajes de programación, instalación de Android Studio	
Actividad sugerida previa	**M1-01, M1-02**	
Tiempo	15 minutos	
Actividad sugerida:	1. Crear tu primer activity 2. En lenguaje de programación JAVA dentro de Android Studio, crea una app vacía. 3. Crear un arreglo con 1 millón de	

<table>
<tr>
<td></td>
<td>elementos enteros de manera aleatoria.
4. Ordenar los elementos.
5. Medir el tiempo de inicio a fin.
6. Imprimir en el logcat los resultados obtenidos.

Subir actividad en el repositorio de github creado por el alumno.
1. android_2025_m1_03.pdf
2. código fuente.</td>
<td></td>
</tr>
<tr>
<td>Criterios de evaluación</td>
<td>
<table>
<tr><td>Criterios</td><td>Puntos máximos</td></tr>
<tr><td>Código fuente solicitado</td><td>9</td></tr>
<tr><td>Presentación, contenido y legibilidad del código</td><td>1</td></tr>
<tr><td>Total</td><td>10</td></tr>
</table>
</td>
</tr>
</table>

Unidad II .Control y versionamiento con Git, Github y Gitflow (opcional)

- ¿Qué es el sistema de control de versiones?
- Instalación de *Git*
- Trabajando con *Git Bash*
- Crear un repositorio en *Git* Comparar
- Cambios de código
- Trabajar con *GitHub*
- Crear un repositorio en *GitHub* Clonar un repositorio desde *GitHub*
- Insertar el código en el repositorio remoto
- Trabajar en equipo
- Cómo un desarrollador envía código a un repositorio remoto
- Trabajar con commits
- Trabajar con branches en *GitHub*
- Resolución de conflictos de *Merge*
- Creación de un pull request (*PR*)
- Revertir cambios del directorio de trabajo

Actividades del Unidad II

- Actividad-M2-01

Tabla de **Actividad-M2-01**		
Actividad	Mi segundo repositorio en Git/Github	

Objetivo del aprendizaje	Conocer las ventajas del sistema de control de versiones con protocolo de seguridad *Secure Shell Protocol* (*SSH*)	
Conocimientos previos requeridos	Sistemas operativos, lenguajes de programación, software, sistema, seguridad.	
Actividad sugerida previa	**N/A**	
Requisitos necesarios	**Cuenta en Github.com**	
Tiempo	45 minutos	
Descripción de la actividad	1. Generar un par de llaves SSH 2. Subir llave pública de SSH a su cuenta de *Github* 3. Crear un segundo repositorio público en github.com 4. Crear un rama bajo el nombre “M2-01” 5. En el directorio raíz subir un archivo en blanco bajo el nombre M2-01.txt 6. Hacer un *commit* con su nombre completo. Nombre_apellidoPaterno_apellidoMaterno (ej cristian_cardoso_arellano) 7. Hacer *push* de los cambios 8. Subir un sólo archivo en formato PDF en el repositorio original de github creado por el alumno.	

	android_2025_m2_01.pdf	
Notas de ayuda	Realizar todos los niveles de https://learngitbranching.js.org/?locale=es_MX	
Criterios de evaluación		

Criterios	Puntos máximos
Repositorio en *Github*	1
Rama creada	1
Ejecución por *SSH*	3
Commit generado	3
Presencia del archivo M2-01.txt	2
Total	10

Unidad III. Aprender sobre variables, constantes, cadenas y operadores
• Constantes y variables

- Definición de constantes y variables
- Tipo safety
- Tipos numéricos
- Tipos enteros
- Valores dobles y de coma flotante
- El tipo booleano
- El tipo de cadena
- Enums
- Operadores
- El operador de asignación
- Operadores de comparación
- Operadores aritméticos
- El operador remainder
- Operadores de asignación compuestos
- El operador condicional ternario
- El operador lógico NOT
- El operador lógico *AND*
- El operador lógico *OR*

Actividades del Unidad III

- Actividad-M3-01

Tabla de **Actividad-M3-01**		
Actividad	Mi primer calculadora en *Android Studio*	

Objetivo del aprendizaje	Comprender y practicar los operadores básicos del lenguaje *Java*	
Conocimientos Criterios requeridos	Sistemas operativos, lenguajes de programación, instalación de *Java*	
Actividad sugerida previa	**M1-01**	
Tiempo	45 minutos	
Descripción de la actividad	1. Realiza un programa de tipo calculadora sin interfaz gráfica, la calculadora deberá de incluir las 4 operaciones básicas (suma, resta, multiplicación, división) 2. Creación de interfaz calculadora. Nota: interfaces de programación. https://docs.oracle.com/javase/tutorial/java/IandI/createinterface.html 3. El resultado de las operaciones deberá de desplegarse en la salida estándar del logcat. 4. Deberá utilizar los conceptos revisados en la **Unidad** 3. 5. La calculadora deberá de validar NaN 6. Puede utilizar las clases bases de: https://github.com/CristianCardosoA/ANDROID-2696/blob/main/calc.zip 7. Subir un sólo archivo en formato PDF en el repositorio de github creado por el alumno. 1. android_2025_m3_01.pdf	

	2. código fuente.	
Criterios de evaluación		

Criterios	Puntos máximos
Estructura (abstracción, modularización) del código de la calculadora	5
Función de suma. resta, multiplicación	2
Función de división	1
Manejo de error NaN	1
Presentación y	1

contenido	
Total	10

Unidad IV. Activities
● Ciclo de vida de una actividad en Android Studio. ● Definición de vistas utilizando XML. ● Unión entre código de programación JAVA y XML con Android Studio. ● Layouts simples.
Actividades del Unidad IV
● Actividad-M4-01

Tabla de **Actividad-M4-01**

Actividad	Ciclo de vida de una *activity*	
Objetivo del aprendizaje	Comprender el ciclo de vida de una actividad.	
Conocimientos previos requeridos	Sistemas operativos, lenguajes de programación, instalación de Android Studio	
Actividad sugerida previa	**M1-01**	
Tiempo	45 minutos	
Descripción de la actividad	1. Crea una *activity* simple en Android Studio en lenguaje de programación Java. 2. Imprimir un mensaje personalizado en cada uno de los métodos del ciclo de vida de una actividad. ● onCreate() ● onStart() ● onPause() ● onRestart() ● onResume() ● onStop() ● onDestroy() Subir actividad en el repositorio de github creado por el alumno. 1. android_2025_m4_01.pdf 2. código fuente.	

Criterios de evaluación	

Criterios	Puntos máximos
Código fuente	8
Presentación y contenido	2
Total	10

Unidad V. Control de flujo, funciones y *lambdas*

- Declaraciones condicionales
- La sentencia *if*
- Ejecución de código condicional con la instrucción if...else
- La declaración *switch*
- El bucle *for-in*
- Uso del bucle *for-in*
- El ciclo while
- Usando el ciclo while
- Usando el ciclo de repetición *while*
- Uso de sentencias case y *where* con sentencias condicionales y bucles
- Filtrado con la sentencia *where*
- Filtrado con la instrucción *for-case*
- Usando la sentencia *if-case*
- Declaraciones de transferencia de control
- La declaración de *break*

- Declaraciones condicionales
- La sentencia *if*
- Uso de una función de parámetro único
- Uso de una función multiparámetro
- Definición de los valores predeterminados de un parámetro
- Agregar nombres de parámetros externos
- Uso de parámetros variádicos
- Parámetros de entrada y salida
- Una introducción a los *lambdas*
- Sintaxis abreviada para *lambdas*

Actividades del Unidad V

- Actividad-M5-01
- Actividad-M5-02

Tabla de **Actividad-M5-01**		
Actividad	Estructuras con la sentencia *For*	
Objetivo del aprendizaje	Comprender y practicar la estructura de control *For*	
Conocimientos previos	Sistemas operativos, lenguajes de programación, instalación de Android Studio	

requeridos		
Actividad sugerida previa	**N/A**	
Tiempo	45 minutos	Material de apoyo
		Lecture 19: Dynamic Programming I: Fibonacci, Shortest Paths https://www.youtube.com/watch?v=OQ5jsbhAv_M&t=1244s
Descripción de la actividad	1. Calcula la secuencia de fibonacci de cualquier número "*n*" 2. Deberá de emplear la estructura de control "*For*" o "Recursividad" para la ejecución de su programa. 3. Agregar dos botones en la vista, utilizando recursos XML. a. Primer botón de avance b. Segundo botón de retroceso. 4. Agregar un TextView en la vista utilizando recursos XML, éste *textView* mostrará el valor actual de la secuencia *fibonacci*. 5. Al dar click en el botón deberá de realizar la secuencia *fibonacci* empezando con el número 1. #1 click, el textview deberá de mostrar el valor de: 1 #2 click, el textview deberá de mostrar el valor de: 1	

<table>
<tr><td></td><td>#3 click, el textview deberá de mostrar el valor de: 2
#4 click, el textview deberá de mostrar el valor de: 3

#5 click, el textview deberá de mostrar el valor de: 5
#6 click, el textview deberá de mostrar el valor de: 8

… consecutivamente …

Incluir “memoization” en el cálculo de la secuencia fibonacci.

Subir actividad en el repositorio de github creado por el alumno.
1. android_2025_m5_01.pdf
2. código fuente.</td><td></td></tr>
<tr><td>Criterios de evaluación</td><td colspan="2"><table>
<tr><td>Criterios</td><td>Puntos máximos</td></tr>
<tr><td>Uso de la estructura for para resolución del problema</td><td>10</td></tr>
<tr><td>Total</td><td>10</td></tr>
</table></td></tr>
</table>

Tabla de **Actividad-M5-02**		
Actividad	Conteo de calorías	
Objetivo del aprendizaje	Comprender y practicar la estructura de control	
Conocimientos previOS© requeridos	Sistemas operativos, lenguajes de programación, instalación de Java	
Actividad sugerida previa	**N/A**	
Tiempo	90 minutos	
Descripción de la actividad	1. A partir del siguiente problema, resolver la solución en lenguaje Java en Android	

	Studio, imprimir el resultado en el logcat. *Descripción del problema* *--- Conteo de calorías ---* *Los renos de Papá Noel suelen comer comida normal para renos, pero necesitan mucha energía mágica para entregar regalos en Navidad. Por eso, su bocadillo favorito es un tipo especial de carambola que solo crece en lo profundo de la jungla. Los Elfos te han traído en su expedición anual al bosque donde crece la fruta.* *Para suministrar suficiente energía mágica, la expedición debe recuperar un mínimo de cincuenta estrellas antes del 25 de diciembre. Aunque los Elfos te aseguran que la arboleda tiene mucha fruta, decides agarrar cualquier fruta que veas por el camino, por si acaso.* *Recoge estrellas resolviendo acertijos. Habrá dos rompecabezas disponibles cada día en el calendario de Adviento; el segundo rompecabezas se desbloquea cuando completas el primero. Cada rompecabezas otorga una estrella. ¡Buena suerte!* *La jungla debe estar demasiado cubierta de maleza y ser difícil de transitar en vehículos o acceder desde el aire; La expedición de los Elfos tradicionalmente va a pie. A medida que sus barcos se acercan a tierra, los Elfos comienzan a hacer un inventario de sus suministros. Una consideración importante es la comida, en*	

	particular, la cantidad de calorías que lleva cada Elfo (su entrada de rompecabezas). *Los Elfos se turnan para anotar el número de Calorías que contienen las distintas comidas, meriendas, raciones, etc. que han traído consigo, un elemento por línea. Cada Elfo separa su propio inventario del inventario del Elfo anterior (si lo hay) con una línea en blanco.* *Por ejemplo, supongamos que los Elfos terminan de escribir las Calorías de sus artículos y terminan con la siguiente lista:* *1000* *2000* *3000* *4000* *5000* *6000* *7000* *8000* *9000* *10000* *Esta lista representa las Calorías de la comida que llevan cinco Elfos:* *El primer Elfo lleva comida con 1000, 2000 y 3000 Calorías, un total de 6000 Calorías.*	

	El segundo elfo lleva un alimento con 4000 calorías. *El tercer Elfo lleva comida con 5000 y 6000 Calorías, un total de 11000 Calorías.* *El cuarto Elfo lleva comida con 7000, 8000 y 9000 Calorías, un total de 24000 Calorías.* *El quinto Elfo lleva un alimento con 10000 Calorías.* *En caso de que los Elfos tengan hambre y necesiten refrigerios adicionales, necesitan saber a qué Elfo preguntar: les gustaría saber cuántas Calorías lleva el Elfo que lleva la mayor cantidad de Calorías. En el ejemplo anterior, esto es 24000 (llevado por el cuarto Elfo).* *Encuentra al duende que lleva más calorías.* 2. *Pregunta a resolver ¿Cuántas calorías totales lleva ese elfo?* 3. *Input data: https://github.com/CristianCardosoA/FCA-iOS©-2023-1/blob/main/inputs/input-M5-02* 4. Subir actividad en el repositorio de github creado por el alumno. 1. android_2025_m5_02.pdf 2. código fuente.	

Criterios de evaluación		

Criterios	Puntos máximos
El resultado correcto	10
Total	10

Unidad VI. Uso de colecciones *en Java con Android Studio*

- Tipos de colecciones *Java*
- Arreglos
- Creación e inicialización de matrices
- Accediendo al elemento de la matriz
- Contar los elementos de un arreglo
- Agregar a una matriz
- Insertar un valor en una matriz
- Sustitución de elementos en una matriz
- Quitar elementos de una matriz
- Fusionando dos arreglos
- Recuperar un subarreglo
- Hacer cambios masivos en una matriz
- Algoritmos para arreglos
- Iterando sobre una matriz
- Diccionarios
- Creación e inicialización de Hashmap
- Acceder a los valores del Hashmap
- Contar las claves o valores en un Hashmap

- Actualización del valor de un key
- Agregar un par clave-valor
- Eliminar un par clave-valor
- Establecer
- Inicializar un conjunto
- Insertar elementos en un conjunto
- Determinar el número de elementos en un conjunto
- Comprobar si un conjunto contiene un elemento
- Iterando sobre un conjunto
- Eliminar elementos de un conjunto
- Establecer operaciones

Actividades del Unidad VI

- Actividad-M6-01

Tabla de **Actividad-M6-01**		
Actividad	Central de aviones	
Objetivo del	Practicar estructura de datos en *Java*	

aprendizaje		
Conocimientos previos requeridos	Programación orientada a objetos, estructura de datos	
Actividad sugerida sugerida previa	M1-01	
Tiempo	12 hr	
Puntos adicionales	El desarrollo efectivo de la actividad ofrece 5 puntos sobre el examen parcial.	
Descripción de la actividad	1. Realiza un algoritmo de la solución para la resolución del problema “Colisión de aviones en un aeropuerto” 2. Ejemplo visual del problema: https://xorzy.github.io/coc/airplane/ 3. Los casos de pruebas serán proporcionados por el profesor durante clase. 4. Descargar el proyecto base en https://github.com/CristianCardosoA/AndroidFCA24-1/tree/main/Aviones 5. Deberá tener las siguientes funcionalidades: ● Grid visible. ● Número de pasos. ● Número de colisiones. ● Aviones ● Dirección del avión. ● Colisión.	

<table>
<tr><td></td><td>

- Botón de paso siguiente.
- Botón de paso anterior.

6. Subir actividad en el repositorio de github creado por el alumno.
1. android_2025_m5_02.pdf
2. código fuente.

</td><td></td></tr>
<tr><td>Criterios de evaluación</td><td>

Criterios	Puntos máximos
Funcionalidad	8
UI	1
Resolución de problema en código	1
Total	10

</td><td></td></tr>
</table>

Temas adicionales/opcionales para descubrimiento profesional

- Patrones de diseño en Android
- Persistencia en Android©
- Comunicación HTTP para consumir servicios REST
- Data Binding for Android©
- Despliegue en tienda
- Automatización
- Unit Testing

Tabla de **Actividad Optativa 01**		
Actividad	*Road map* Android DEV Professional	
Objetivo del aprendizaje	Identificar áreas de oportunidad de un profesional encargado en desarrollar, diseñar aplicaciones Android empresariales	
Tiempo	1 hr	
Descripción de la actividad	1. Realiza un documento que menciona la matriz de habilidad para un *Android Developer Skill Matrix*	

Tabla de **Actividad Optativa 02**		
Actividad	Explorando *Http web services*	
Objetivo del aprendizaje	Comprender que son los *REST* servicios web, y qué relación tienen con los aplicativos Android	
Conocimientos prev requeridos	Base datos informática y sistemas operativos	
Tiempo	1 hr	
Descripción de la actividad	1. Investiga que es un *REST* web service y menciona que relación tiene con un aplicativos *Android* 2. Realiza un documento ilustrativo que refleja un panorama general sobre el consumo de los s web en *Android* 3. Desarrolla una app sencilla donde se consuma un servicio rest desde *https://api.publicapis.org* - */Get /Entries* 4. La app deberá desplegar en pantalla el resultado de la operación *http get*	

EXÁMENES

De acuerdo con los lineamientos del modelo educativo, tienes tres períodos a lo largo del semestre para presentar tus exámenes parciales (consulta las fechas en el calendario de inscripción a parciales y globales en el Portal SUAyED), tú decides el período en el que los realizarás. Si tu asignatura es **optativa**, deberás consultar los períodos y número de exámenes con tu asesor.

Para esta asignatura están programados de la siguiente manera:

- **Exámenes Parciales:**

Deberás entregar las actividades de aprendizaje de las unidades implicadas en cada parcial, **antes de que inicie el periodo de aplicación, si las entregas durante la aplicación del examen se consideran extemporáneas**. Es importante que te inscribas en cada periodo y cumplas con los lineamientos para su presentación.

NÚMERO	UNIDADES (que lo integran)	VALOR (núm. enteros)
1ro.	1 a 6 **Se debe presentar en el tercer periodo de aplicación de exámenes.**	50%

- **Global. Examen único**

Valor	Requisitos	Aplicación de global
100% de la calificación final	Ninguno	6,7 y del 9 al 12 de junio de 2025

PORCENTAJES Y ESCALA DE EVALUACIÓN Y ACREDITACIÓN

Concepto	Porcentajes
Actividades de aprendizaje	50 %
Exámenes parciales	50 %
Total	**100 %**

- **Escala de evaluación:**

Rango	Calificación
1.00 a 5.99	**5**
6.00 a 6.59	**6**
6.60 a 7.59	**7**
7.60 a 8.59	**8**
8.60 a 9.59	**9**
9.60 a 10.00	**10**

FUNCIONES DEL ASESOR

Por apoyar tu proceso de aprendizaje autónomo, el asesor tiene las siguientes funciones:

1. Apoyar y guiar en la resolución de dudas y desarrollo de actividades; a través de los canales de comunicación oficiales.
2. Calificar y retroalimentar las actividades en plataforma educativa en un lapso no mayor a ocho días hábiles después de la fecha de entrega establecida en el calendario.
3. Recomendar recursos didácticos para ampliar tu conocimiento. No es su obligación facilitarte: copias, libros, archivos digitales o proporcionar ligas directas de la BIDI.
4. Enviar las calificaciones al finalizar el semestre de manera personalizada por correo electrónico.

DATOS DEL ASESOR O GRUPO DE ASESORES

Nombre	Correo electrónico
L.I. Cristian Cardoso Arellano	cris52@comunidad.unam.mx

MEDIOS DE COMUNICACIÓN SUGERIDOS CON EL ASESOR
El alumno podrá comunicarse con el profesor en cualquiera de los siguientes medios para la aclaración de dudas o comentarios relacionados a la asignatura y/o en caso de alguna eventualidad externa. ● Slack© ○ Invitación: https://join.slack.com/t/fca-unam-info/shared_invite/zt-2k23kfsq9-yBsKeCVVKGtcZuODmcTO4g ● Correo electrónico ○ cris52@comunidad.unam.mx Se le sugiere al alumno asistir a las asesorías presenciales en el aula asignada en el periodo 2025-2 en el edificio de *SUAYED* dentro de la Facultad de Contaduría y Administración en un horario de 20:00 p.m. a 22:00 p.m. lunes y viernes.

Enseñar no es transferir conocimiento, sino crear las posibilidades para su propia producción o construcción.
Paulo Freire