



## Anexo 2

### Pruebas para la obtención de títulos de Técnico y Técnico Superior

#### Convocatoria correspondiente al curso académico 2023-2024

(Resolución de 29 de diciembre de 2023 de la Dirección General de Educación Secundaria, Formación Profesional y Régimen Especial)

DATOS DEL ASPIRANTE			FIRMA
Apellidos:			
Nombre:	D.N.I. N.I.E. o Pasaporte:	Fecha:	

Código del ciclo:	Denominación completa del título:
IFCS02	Técnico Superior en Desarrollo de Aplicaciones Multiplataforma
Clave o código del módulo:	Denominación completa del módulo profesional:
489	Programación multimedia y dispositivos móviles

INSTRUCCIONES GENERALES PARA LA REALIZACIÓN DE LA PRUEBA
<ul style="list-style-type: none"><li>El examen tendrá una duración de 2 horas.</li><li>La prueba consta de un examen tipo test con cuatro opciones de las cuales solamente una es correcta.</li><li>Cada pregunta se responderá en el espacio dejado al efecto en la hoja de respuestas. Se rellenarán los recuadros para señalar la respuesta seleccionada.</li><li>Si se quiere rectificar una respuesta contestada, se borrará toda la casilla de la respuesta incorrecta con Tipp-Ex o corrector, tal y como se puede apreciar en el siguiente ejemplo:<ul style="list-style-type: none"><li>Se elimina la selección de <b>b</b> para seleccionar la opción <b>d</b>: <input type="checkbox"/>a <input type="checkbox"/>b <input type="checkbox"/>c <input checked="" type="checkbox"/>d</li></ul></li><li>Se dispondrá de una hoja para borrador (o de varias si se requieren), que será proporcionada por el centro. Esa hoja se entregará obligatoriamente al final junto con el examen, si bien nada de lo escrito en la hoja de borrador se valorará en la corrección.</li><li>Sólo se utilizará bolígrafo negro o azul y Tipp-Ex o corrector, no permitiéndose usar bolígrafo rojo, lapicero, etcétera.</li><li>No se podrá emplear ningún dispositivo electrónico.</li><li>Cualquier tachadura o borrón en una respuesta podrá invalidar toda la puntuación de la pregunta asociada.</li></ul>

CRITERIOS DE CALIFICACIÓN Y VALORACIÓN
<ul style="list-style-type: none"><li>El test se calificará sobre 10 puntos. Todas las preguntas se calificarán equitativamente con la misma cantidad de puntos. En cada pregunta se plantearán varias respuestas, y se deberá señalar la única que se considere correcta, según el caso. Cada respuesta correcta que se marque se valorará con 0,25 puntos, y si se marca alguna incorrecta, se valorará con una cantidad negativa equivalente a 0,075 puntos, es decir, se descontarán 0,075 puntos. Si no se está seguro de si una respuesta es correcta o no, y no se marca, no sumará ni restará puntos.</li><li>Calificación final del módulo profesional:<ul style="list-style-type: none"><li>El alumno obtendrá en el módulo profesional una calificación entera entre 1 y 10.</li><li>Dicha calificación se calculará:<ul style="list-style-type: none"><li>Si la calificación conseguida en la prueba es inferior a 5: se truncará dicha calificación.</li><li>Si la calificación conseguida en la prueba es igual o superior a 5 y los decimales:<ul style="list-style-type: none"><li>Son inferiores a 0'5: se redondeará al entero inferior más próximo.</li><li>Son iguales o superiores a 0'5: se redondeará al entero superior más cercano.</li></ul></li></ul></li><li>La anterior regla tiene una excepción: las notas de examen inferiores a 1 se redondearán a 1.</li></ul></li></ul>



Unión Europea

Fondo Social Europeo

"El FSE invierte en tu futuro"



IES Rey Fernando VI



Comunidad de Madrid

## **CONTENIDO DE LA PRUEBA**

**Parte 1: análisis de tecnologías para aplicaciones en dispositivos móviles. Resultados de aprendizaje:**  
**Aplica tecnologías de desarrollo para dispositivos móviles evaluando sus características y capacidades.**

**1) ¿Cuál de las siguientes es una limitación común al ejecutar aplicaciones en dispositivos móviles?**

- a) Tamaño de pantalla reducido.
- b) Mayor capacidad de almacenamiento.
- c) Mayor potencia de procesamiento.
- d) Conexión a Internet estable.

**2) ¿Qué factor puede limitar la ejecución eficiente de aplicaciones en dispositivos móviles?**

- a) La duración de la batería.
- b) La cantidad de RAM disponible.
- c) La velocidad de la conexión Wi-Fi.
- d) La compatibilidad con múltiples idiomas.

**3) ¿Qué tecnología reduce significativamente el consumo de batería en aplicaciones móviles al optimizar la gestión de recursos?**

- a) Procesadores multicore
- b) Uso intensivo de GPS
- c) Aplicaciones en segundo plano
- d) Actualizaciones automáticas en red

**4) ¿Por qué es importante considerar las limitaciones de hardware al desarrollar aplicaciones móviles?**

- a) Para garantizar la seguridad de la aplicación.
- b) Para optimizar el rendimiento y la experiencia del usuario.
- c) Para aumentar los requisitos de almacenamiento.
- d) Para reducir la compatibilidad con diferentes dispositivos.

**5) ¿Qué limitación puede surgir debido a la variedad de dispositivos móviles en el mercado?**

- a) Limitaciones de conectividad Bluetooth.



Unión Europea

Fondo Social Europeo

"El FSE invierte en tu futuro"



IES Rey Fernando VI



Comunidad de Madrid

- b) Incompatibilidad con sistemas operativos específicos.
- c) Limitaciones de tamaño de pantalla uniforme.
- d) Capacidades de procesamiento consistentes.

**6) ¿Cuál de las siguientes tecnologías ayuda a mitigar las limitaciones de rendimiento en dispositivos móviles al permitir la ejecución de código nativo en múltiples plataformas?**

- a) HTML5
- b) Flutter
- c) Swift
- d) Kotlin

**7) ¿Cuál de los siguientes conceptos se centra en optimizar el consumo de energía de una aplicación móvil para maximizar la duración de la batería y minimizar el impacto en el rendimiento del dispositivo?**

- a) Throttling
- b) Profiling
- c) Minificación
- d) Ninguna de las anteriores

**8) ¿Qué aspecto de los dispositivos móviles puede afectar significativamente la experiencia del usuario al ejecutar aplicaciones complejas?**

- A) Tamaño de pantalla grande
- B) Conectividad de red inestable
- C) Disponibilidad de aplicaciones multiplataforma
- D) Facilidad de uso de la interfaz de usuario

**Parte 2: programación de aplicaciones para dispositivos móviles. Resultados de aprendizaje: Desarrolla aplicaciones para dispositivos móviles analizando y empleando las tecnologías y librerías específicas.**

**Para el siguiente código en Android Kotlin:**

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.EditText
import android.textEditable
```



```
import android.text.TextWatcher
```

```
class DynamicKeyboardActivity : AppCompatActivity() {  
    private lateinit var editText: EditText  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_dynamic_keyboard)  
  
        editText = findViewById(R.id.editText)  
        editText.addTextChangedListener(object : TextWatcher {  
            override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {}  
  
            override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {  
                updateKeyboardLayout(s.toString())  
            }  
  
            override fun afterTextChanged(s: Editable?) {}  
        })  
    }  
  
    private fun updateKeyboardLayout(text: String) {  
  
    }  
}
```

9) ¿Qué método se utiliza para agregar un listener que reaccione a los cambios de texto?

- A) setTextListener
- B) addTextChangedListener
- C) onTextChange
- D) setKeyListener

10) ¿Qué interfaz de Kotlin se implementa para detectar y responder a los cambios de texto?

- A) KeyListener
- B) TextWatcher
- C) TextEditor
- D) EditTextWatcher



**11) En el contexto de MVC, ¿qué componente sería DynamicKeyboardActivity?**

- A) Modelo
- B) Vista
- C) Controlador
- D) Servicio

**12) ¿Qué se debe implementar adicionalmente para que updateKeyboardLayout modifique efectivamente el teclado?**

- A) Una API externa
- B) Un InputMethodManager
- C) Una base de datos local
- D) Un BroadcastReceiver

**13) ¿Cómo se podría mejorar la implementación de TextWatcher para optimizar el rendimiento en dispositivos con menos recursos?**

- A) Usando lateinit para todos los componentes.
- B) Minimizando el número de llamadas a updateKeyboardLayout.
- C) Eliminando el método onTextChanged.
- D) Implementando TextWatcher en una clase aparte.

**14) ¿Qué patrón de diseño se recomienda para implementar la comunicación entre la vista y el modelo en aplicaciones móviles, garantizando al mismo tiempo la separación de preocupaciones?**

- A) Singleton
- B) Factory
- C) Model-View-Controller (MVC)
- D) Model-View-ViewModel (MVVM)

**15) ¿Qué mecanismo utiliza Android para permitir que una aplicación ejecute un proceso en segundo plano, incluso cuando la aplicación no está en primer plano?**

- A) Intents
- B) Services



Unión Europea  
Fondo Social Europeo  
"El FSE invierte en tu futuro"



Comunidad de Madrid

- C) AsyncTask
- D) Broadcast Receivers

**16) ¿Qué archivo en una aplicación Android especifica los componentes de la aplicación, como actividades, servicios, y permisos de usuario?**

- A) MainActivity.java
- B) AndroidManifest.xml
- C) app.gradle
- D) index.js

**Parte 3: utilización de librerías multimedia integradas. Resultados de aprendizaje: Desarrolla programas que integran contenidos multimedia analizando y empleando las tecnologías y librerías específicas.**

**Build.gradle:**

```
dependencies {  
    // CameraX core library  
    implementation "androidx.camera:camera-camera2:1.0.0-beta07"  
    implementation "androidx.camera:camera-lifecycle:1.0.0-beta07"  
    implementation "androidx.camera:camera-view:1.0.0-alpha20"  
}
```

**activity\_main.xml**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
  
    <androidx.camera.view.PreviewView  
        android:id="@+id/viewFinder"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent" />  
  
    <ImageView  
        android:id="@+id/imgResult"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:visibility="gone" />  
</RelativeLayout>
```



## MainActivity.kt

```
class MainActivity : AppCompatActivity() {

    private lateinit var imageCapture: ImageCapture

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        startCamera()
    }

    private fun startCamera() {
        val cameraProviderFuture = ProcessCameraProvider.getInstance(this)
        cameraProviderFuture.addListener({
            val cameraProvider: ProcessCameraProvider = cameraProviderFuture.get()
            val preview = Preview.Builder()
                .build()
                .also {
                    it.setSurfaceProvider(findViewById<PreviewView>(R.id.viewFinder).surfaceProvider)
                }

            imageCapture = ImageCapture.Builder().build()

            try {
                cameraProvider.unbindAll()
                cameraProvider.bindToLifecycle(
                    this, CameraSelector.DEFAULT_BACK_CAMERA, preview, imageCapture
                )
            } catch (exc: Exception) {
                // Handle exception
            }
        }, ContextCompat.getMainExecutor(this))
    }

    private fun takePhoto() {
        val photoFile = File.createTempFile("photo_", ".jpg", externalCacheDir)
        val outputOptions = ImageCapture.OutputFileOptions.Builder(photoFile).build()

        imageCapture.takePicture(
            outputOptions,
```



```
ContextCompat.getMainExecutor(this),
object : ImageCapture.OnImageSavedCallback {
    override fun onError(exc: ImageCaptureException) {
        // Handle error
    }

    override fun onImageSaved(output: ImageCapture.OutputFileResults) {
        val bitmap = BitmapFactory.decodeFile(photoFile.absolutePath)
        val grayBitmap =
            Bitmap.createBitmap(bitmap.width, bitmap.height, Bitmap.Config.ARGB_8888)
        val canvas = Canvas(grayBitmap)
        val paint = Paint()
        val colorMatrix = android.graphics.ColorMatrix()
        colorMatrix.setSaturation(0f)
        paint.colorFilter = android.graphics.ColorMatrixColorFilter(colorMatrix)
        canvas.drawBitmap(bitmap, 0f, 0f, paint)

        runOnUiThread {
            findViewById<ImageView>(R.id.imgResult).apply {
                setImageBitmap(grayBitmap)
                visibility = View.VISIBLE
            }
        }
    }
})
}
```

**17) ¿Cuál es el propósito de usar ProcessCameraProvider en la aplicación de captura de fotos?**

- a) Para aplicar filtros de imagen.
- b) Para capturar audio.
- c) Para vincular el ciclo de vida de la cámara con el de una Activity o Fragment.
- d) Para crear un archivo de imagen temporal.

**18) ¿Qué función realiza la clase Preview.Builder() en el código proporcionado?**

- A) Configura y proporciona una vista previa de la imagen a capturar.
- B) Crea y almacena imágenes fijas.





Unión Europea  
Fondo Social Europeo  
"El FSE invierte en tu futuro"



Comunidad de Madrid

C) Configura la conexión de datos de la cámara.

D) Captura imágenes en video.

**19) Al aplicar un filtro de escala de grises a una imagen, ¿qué objeto se utiliza para cambiar la saturación a 0?**

- a) Paint
- b) BitmapFactory
- c) ColorMatrix
- d) Canvas

**20) ¿Cuál de las siguientes afirmaciones es verdadera sobre la configuración de ImageCapture para capturar fotos?**

- a) Se requiere un ImageCapture.Builder para especificar opciones de captura avanzadas.
- b) ImageCapture automáticamente guarda las imágenes en la galería del dispositivo.
- c) ImageCapture utiliza OpenGL ES para el procesamiento de imágenes.
- d) La clase ImageCapture es parte de la biblioteca Freetype.

**21) ¿Cómo se muestra la imagen con filtro aplicado en la UI de la aplicación?**

- a) Cambiando la visibilidad de PreviewView a View.GONE.
- b) Utilizando ImageView.setImageBitmap() con el bitmap filtrado.
- c) Aplicando un ColorMatrixColorFilter directamente en el PreviewView.
- d) Creando un nuevo ImageView en tiempo de ejecución.

**22) ¿Cuál es el propósito del método runOnUiThread en el contexto de actualizar la interfaz de usuario con la imagen filtrada?**

- a) Para aplicar filtros de imagen de forma asíncrona.
- b) Para garantizar que la actualización de la interfaz de usuario se realice en el hilo principal de la aplicación.
- c) Para iniciar un nuevo hilo que se encargue de la renderización de imágenes.
- d) Para retrasar la ejecución del código hasta que la imagen esté completamente cargada.

**23) ¿Cómo se maneja un error durante el proceso de toma de fotografías en el código proporcionado?**



- A) Reiniciando la cámara automáticamente.
- B) Lanzando una excepción que debe ser capturada por el método llamado.
- C) Ejecutando un bloque de código específico dentro del método onError.
- D) Devolviendo un valor de error al método principal.

**24) ¿Qué hace setSurfaceProvider() dentro del contexto de configuración de la cámara?**

- A) Configura el proveedor de almacenamiento para las imágenes.
- B) Establece el proveedor de la superficie de visualización para la previsualización.
- C) Define el proveedor de servicios de la cámara.
- D) Ajusta las propiedades de conexión de la cámara.

**Parte 4: análisis de motores de juegos. Resultados de aprendizaje: Selecciona y prueba motores de juegos analizando la arquitectura de juegos 2D y 3D.**

**25) ¿Qué técnica de animación es comúnmente utilizada en juegos para dispositivos móviles para animar personajes y objetos?**

- a) Captura de movimiento
- b) Animación por cuadros clave (keyframe animation)
- c) Animación vectorial
- d) Stop motion

**26) En la arquitectura de un juego para dispositivos móviles, ¿qué componente es responsable de gestionar el estado del juego, la transición entre pantallas y la lógica del juego?**

- a) Gestor de servicios
- b) Enrutador de pantallas
- c) Gestor de escenas
- d) Interfaz de usuario

**27) ¿Qué componente de un motor de juegos es crítico para determinar cómo los objetos interactúan visualmente entre sí y con el entorno del juego en un espacio tridimensional?**

- a) Motor gráfico o de renderizado
- b) Grafo de escena



- c) Detector de colisiones
- d) Motor de físicas

**28) Cuando se aplican modificaciones sobre juegos existentes en dispositivos móviles, ¿qué aspecto es crucial para asegurar una buena experiencia de usuario?**

- a) Aumentar la dificultad del juego
- b) Mejorar la interfaz de usuario y la usabilidad
- c) Cambiar la banda sonora original por una nueva
- d) Disminuir la resolución de los gráficos

**29) Para implementar la detección de colisiones en un juego desarrollado con libGDX, ¿qué biblioteca externa es comúnmente utilizada?**

- a) Scene2D
- b) Box2D
- c) Bullet Physics
- d) LibGDX Collider

**Parte 5: desarrollo de juegos 2D y 3D. Resultados de aprendizaje: Desarrolla juegos 2D y 3D sencillos utilizando motores de juegos.**

**Juego físico usando Kotlin y Libgdx**

```
class PhysicsGame : ApplicationAdapter() {  
    private lateinit var camera: OrthographicCamera  
    private lateinit var world: World  
    private lateinit var debugRenderer: Box2DDebugRenderer  
  
    override fun create() {  
        camera = OrthographicCamera(Gdx.graphics.width.toFloat(), Gdx.graphics.height.toFloat())  
        world = World(0f, -10f, true)  
        debugRenderer = Box2DDebugRenderer()  
  
        // Crear suelo  
        val bodyDef = BodyDef()  
        bodyDef.position.set(0f, 0f)  
        val body = world.createBody(bodyDef)
```



```
val groundShape = PolygonShape()
groundShape.setAsBox(camera.viewportWidth, 1f)
```

```
val fixtureDef = FixtureDef()
fixtureDef.shape = groundShape
body.createFixture(fixtureDef)
groundShape.dispose()
```

*// Crear bola*

```
bodyDef.type = BodyDef.BodyType.DynamicBody
bodyDef.position.set(camera.viewportWidth / 2, camera.viewportHeight)
val ballBody = world.createBody(bodyDef)
```

```
val ballShape = PolygonShape()
ballShape.setAsBox(1f, 1f)
```

```
fixtureDef.shape = ballShape
fixtureDef.density = 1f
fixtureDef.restitution = 0.5f // Hace que la bola rebote
ballBody.createFixture(fixtureDef)
ballShape.dispose()
```

```
}
```

```
override fun render() {
    Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT)
    camera.update()
    debugRenderer.render(world, camera.combined)
    world.step(1/60f, 6, 2)
}
```

```
override fun dispose() {
    world.dispose()
    debugRenderer.dispose()
}
}
```

**30) ¿Cuál es la función de OrthographicCamera en el contexto del programa proporcionado?**

- A) Manejar la física del juego.
- B) Proveer una vista 2D sin perspectiva del mundo.
- C) Manipular imágenes
- D) Gestionar la posición de la pantalla en relación a la cámara.



**31) ¿Qué representa el valor -10f en la creación del objeto World?**

- A) La densidad del aire en el mundo.
- B) La velocidad inicial de la bola.
- C) La gravedad que afecta a todos los cuerpos.
- D) La resistencia del aire.

**32) ¿Qué realiza el método world.step en el método render?**

- A) Inicializa los componentes gráficos del mundo.
- B) Actualiza la posición y física de todos los cuerpos en el mundo.
- C) Reinicia el estado del mundo a su configuración inicial.
- D) Guarda el estado actual del juego.

**33) ¿Cómo afecta fixtureDef.density al comportamiento físico del objeto en el juego?**

- A) Cambia la velocidad a la que el objeto se mueve.
- B) Determina la masa del objeto basada en su volumen.
- C) Define la transparencia del objeto.
- D) Establece el grado de reflexión de la luz en el objeto.

**34) ¿Qué define el BodyDef en Box2D?**

- A) El renderizado de un objeto
- B) Las propiedades físicas de un objeto
- C) La textura de un objeto
- D) La posición inicial de un objeto

**35) ¿Qué efecto tiene la propiedad restitution en un objeto de Box2D?**

- A) Controla la velocidad del objeto
- B) Determina la masa del objeto
- C) Afecta cuánto rebota el objeto
- D) Cambia el color del objeto



## Tenemos el siguiente código de un videojuego en Kotlin y libGDX

```
class SensorGame : ApplicationAdapter() {  
    private lateinit var batch: SpriteBatch  
    private lateinit var shapeRenderer: ShapeRenderer  
    private lateinit var viewport: Viewport  
    private lateinit var texture: Texture  
    private var position = Vector3()  
    private var screenSize = Vector3()  
  
    override fun create() {  
        screenSize.set(Gdx.graphics.width.toFloat(), Gdx.graphics.height.toFloat(), 0f)  
        batch = SpriteBatch()  
        shapeRenderer = ShapeRenderer()  
        viewport = FitViewport(800f, 600f, camera)  
        texture = Texture("ball.png") // Asumiendo que existe un recurso "ball.png"  
  
        // Iniciar detección de sensor  
        Gdx.input.isPeripheralAvailable(Gdx.input.Peripheral.Accelerometer)  
    }  
  
    fun update() {  
        val accelX = Gdx.input.accelerometerX  
        position.x -= accelX  
        position.x = Math.max(0f, Math.min(screenSize.x, position.x))  
    }  
  
    override fun render() {  
        update() // Actualiza la lógica del juego  
  
        Gdx.gl.glClearColor(0f, 0f, 0f, 1f)  
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT)  
  
        batch.begin()  
        batch.draw(texture, position.x, position.y)  
        batch.end()  
    }  
  
    override fun dispose() {  
        batch.dispose()  
        texture.dispose()  
        shapeRenderer.dispose()  
    }  
}
```



```
}  
}
```

**36) ¿Qué método asegura que la posición x del sprite se mantenga dentro de los límites de la pantalla?**

- A) viewport.update()
- B) Math.max(0f, Math.min(screenSize.x, position.x))
- C) batch.begin()
- D) Gdx.input.accelerometerX

**37) ¿Para qué se utiliza el Viewport en este contexto?**

- A) Para detectar la inclinación del dispositivo.
- B) Para controlar la ejecución del juego.
- C) Para administrar cómo se proyectan los gráficos en diferentes pantallas.
- D) Para cargar texturas.

**38) ¿Cómo afecta la lectura del acelerómetro (accelX) a la posición del sprite en el juego?**

- A) Cambia la posición y en el eje Y.
- B) Modifica la orientación del sprite.
- C) Altera la posición en el eje X.
- D) Ajusta la escala del sprite.

**39) ¿Cuál es el propósito específico de utilizar Vector3 para almacenar la posición y tamaño de la pantalla en SensorGame?**

- A) Facilitar cálculos tridimensionales en un entorno 2D.
- B) Permitir una representación más precisa de las coordenadas en el espacio.
- C) Aprovechar métodos específicos de Vector3 para manipulación de vectores.
- D) Ninguna de las anteriores.

**40) En el contexto de juegos que utilizan sensores, ¿cuál es una práctica común para suavizar los datos del acelerómetro y evitar saltos bruscos en la entrada?**

- A) Aplicar un filtro de Kalman.



Unión Europea

Fondo Social Europeo

*"El FSE invierte en tu futuro"*



**Comunidad de Madrid**

- B) Usar la media de las últimas N lecturas.
- C) Aplicar un filtro paso bajo.
- D) Incrementar la tasa de muestreo del sensor.