



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



COLEGIO DE
BACHILLERES

EXAMEN PARA LA EVALUACIÓN DE RECUPERACIÓN

EVAREC 2025 - B

PROGRAMACIÓN EN JAVA

GUÍA PARA LA ELABORACION DEL PROYECTO DE PROGRAMACIÓN EN JAVA

CLAVE: 195
PLAN DE ESTUDIOS 2023

DIRECCIÓN DE EVALUACIÓN,
ASUNTOS DEL PROFESORADO
Y ORIENTACIÓN EDUCATIVA

EVAREC-195PJ-25B

Índice

| | |
|---|-----------|
| Índice | 2 |
| Presentación | 3 |
| ¿Cómo usar tu guía?..... | 4 |
| Problemática | 4 |
| Actividades a realizar:..... | 5 |
| Comprobando mis aprendizajes | 11 |
| Bibliografía..... | 14 |

Presentación

Esta guía temática está diseñada para ti, estudiante que te preparas para presentar los mecanismos de regularización, correspondientes al semestre 2025B, en el marco de los programas de estudios de quinto semestre correspondientes al Plan de estudios 2023.

El **currículum laboral** se basa en un enfoque por competencias laborales básicas, con el cual, se busca que el estudiantado obtenga una formación que le permita incorporarse al ámbito laboral y su articulación con los saberes adquiridos en las UAC que pertenecen al currículum fundamental y al ampliado, a fin de que sean capaces de resolver problemas en diversos contextos; sabemos que los mecanismos de regularización representan una oportunidad para demostrar no solo el dominio de los conocimientos adquiridos, sino también la capacidad de superar desafíos académicos y la disposición para alcanzar el éxito educativo; por lo que esta guía ha sido creada con el propósito de acompañarte en tu preparación, proporcionándote una estructura organizada que te ayudará a maximizar tu rendimiento en estas evaluaciones.

Te animamos a abordar esta guía con determinación, aprovechando la oportunidad de consolidar tus conocimientos y mejorar tu comprensión de los contenidos esenciales. Recuerda que la preparación no solo se trata de acumular información, sino de desarrollar habilidades críticas que te servirán a lo largo de tu vida académica y profesional.

¡Éxito en tus estudios!

¿Cómo usar tu guía?

La presente guía contiene una problemática situada que abarca los contenidos de cada uno de los cortes temáticos, esta problemática contiene actividades que deberás desarrollar y te servirán como preparación para presentar tus evaluaciones extraordinarias. Las actividades se derivan de los contenidos esenciales del programa de estudios de la UAC **Programación en Java**, que refieren a lo que debes aprender en el semestre; también se proporcionan recursos digitales en la sección de bibliografía, que apoyan tu proceso de aprendizaje y debes consultarlos para completar las actividades. En el apartado “Comprobando mis aprendizajes” hay una rubrica, asegúrate de consultarla para verificar si cumpliste con lo solicitado. Puedes realizar las actividades a mano o en formato digital; recuerda colocar tu nombre, matrícula, semestre y el nombre de la UAC. Entrega tus actividades a la persona responsable que se designe en tu plantel.

Problemática

Desarrollo de App de Hábitos Saludables

Contexto: En la actualidad, muchas personas enfrentan dificultades para mantener hábitos saludables de manera constante debido al ritmo acelerado de vida, la falta de motivación, y la escasa conciencia sobre la importancia del autocuidado. Actividades esenciales como beber suficiente agua, dormir adecuadamente, realizar ejercicio físico o llevar un control del bienestar emocional suelen ser descuidadas, lo que puede derivar en problemas de salud física y mental a largo plazo.

A pesar de la disponibilidad de información sobre estilos de vida saludables, existe una brecha entre el conocimiento y la acción. Las personas necesitan herramientas prácticas que les ayuden a establecer rutinas, hacer seguimiento de sus progresos y recibir estímulos que los motiven a continuar. En este contexto, la tecnología móvil

representa una oportunidad poderosa para fomentar el autocuidado mediante soluciones accesibles, personalizadas y fáciles de usar.

Solución: La solución planteada es el desarrollo de una **aplicación móvil de hábitos saludables**, que permita al usuario:

- **Registrar hábitos diarios** como tomar agua, dormir bien, hacer ejercicio, entre otros.
- **Visualizar estadísticas** sobre su progreso, mediante gráficos y reportes semanales o mensuales.
- **Establecer metas personalizadas** y recibir notificaciones motivacionales que refuercen el compromiso.
- **Optimizar la experiencia de usuario (UX)** con una interfaz intuitiva y atractiva.

Esta aplicación será desarrollada en Java, utilizando principios de programación orientada a objetos, y siguiendo las fases de planificación, diseño, desarrollo, prueba e implementación. Además, se distribuirá en plataformas comerciales y personales para facilitar su acceso.

Actividades a realizar:

1. **Diagrama UML de clases** que muestre:
 - Nombre de la clase.
 - Atributos con tipo de dato definido.
 - Métodos: visibilidad, tipo de retorno y parámetros.
 - Relaciones entre clases: herencia, polimorfismo, asociación, agregación y composición.
 - Métodos de acceso indicando si los atributos y métodos son públicos, privados o protegidos.
 - Cumple con la nomenclatura UML estándar.
 - Consistencia de nombres y tipos de datos.

2. **Diagrama UML de objetos** que muestre:

- Instancia de la clase identificado con un nombre y su clase correspondiente.
- Atributos con valores asignados al estado del objeto en un momento específico.
- Relaciones entre objetos con líneas de conexión.
- Visibilidad de métodos y atributos.
- Consistencia con el diagrama de clases.
- Cumple con la notación UML estándar.

3. **Diseño de programas básicos de consola o con interfaz en Java que considere:**

- Uso de un IDE: Eclipse, NetBeans, IntelliJ, DEA o cualquier otro.
- Creación del proyecto Java con la estructura de carpetas (src, lib, bin, etc.).
- Código estructurado en clases y métodos.
- Aplicación de principios de modularidad y reutilización de código.
- Uso adecuado de nombres descriptivos para clases, métodos y variables.
- Uso de comentarios explicativos para mejorar la legibilidad (incluyendo Javadoc).
- Declaración correcta de variables y constantes.
- Uso adecuado de tipos de datos y operadores.
- Implementación de estructuras de control de forma eficiente.
- Implementación de excepciones con try-catch y finally que controlen errores comunes de ejecución.
- Algoritmos básicos para el uso de las estructuras de datos.
- Optimización de algoritmos al eliminar redundancias.
- Pruebas de funcionamiento y validación del código.
- Creación de interfaces o interacción con el usuario a través de la consola.
- El resultado de la aplicación directa de la solución programada en el contexto de la problemática.

4. **Elabora un documento que contenga la primera etapa del desarrollo de proyectos (planeación) de una aplicación móvil**, la cual debe contener los siguientes apartados:

- Definición del problema.
- Objetivo general y específicos del proyecto.
- Análisis de mercado sobre el público objetivo.
- Análisis de aplicaciones similares a la que se pretende desarrollar.
- Alcance y limitaciones.
- Lista de requisitos funcionales y no funcionales.
- Justificación de la tecnología de desarrollo (plataforma, lenguaje de programación y sistema gestor de bases de datos relacionales).
- Análisis de factibilidad técnica, operativa, económica y de tiempo.
- Diagrama de Gantt o herramienta de gestión de proyectos.

5. **Elabora Wireframes de baja fidelidad**, considerando:

- Claridad de la estructura o composición que visualice la jerarquía de la información y zonas interactivas y relaciones entre pantallas elaborado a mano o con herramientas digitales.
- Flujo de navegación que muestre de forma gráfica el recorrido que el usuario realiza dentro de la aplicación.
- Enfoque funcional y simple.
- Mockups de alta fidelidad elaborados, considerando:
- Consistencia visual en el uso de colores, tipografías, iconografía y espaciados, así como en el estilo gráfico de las pantallas.
- Aplicación de los principios de desarrollo de interfaces para dispositivos móviles.
- Nivel de detalle y presión en los elementos de navegación, captura de datos y contenido, mostrando una interfaz cercana a la versión final de la aplicación.

- Uso de herramientas digitales como Figma, Justinmind, Adobe XD, InVision o Sketch para su desarrollo.
- Prototipo interactivo para validar navegación y flujo de interacción antes de la programación.
- Optimizar la experiencia de usuario aplicando principios fundamentales de UX que garanticen un flujo de navegación intuitivo, accesibilidad y usabilidad.

6. **Desarrolla la Interfaz gráfica de la aplicación móvil en Android en un IDE,** que considere los siguientes elementos.

1. Proyecto configurado:

- Archivo del proyecto en Android Studio con un layout XML,
- Actividad base seleccionada según la funcionalidad requerida (Empty Activity, Basic Views Activity, Bottom Navigation Views Activity, Responsive Views Activity, entre otras).

2. Diseño UI organizado:

- Archivo activity_main.xml con el contenedor principal establecido (LinearLayout, ConstraintLayout, RelativeLayout),
- Activity_main.xml cuenta con una organización jerarquía clara y con una disposición lógica de los elementos,
- Nombres significativos y descriptivos a los ID de los componentes XML.

3. Interfaz funcional con elementos visuales distribuidos y estilizados:

- Márgenes y padding aplicados adecuadamente para brindar una mejora visual de los elementos que lo integran,

- Elementos gráficos esenciales (TextView, Button, ImageView, EditText, etc.) comunican y se encuentran distribuidos considerando las técnicas visuales,
- Elementos en la interfaz están alineados y distribuidos correctamente,
- Textos en TextView cuentan con tamaño, color y fuente adecuados,
- Botones y elementos interactivos contienen colores, contrastantes y tamaños accesibles para la interacción.

4. El diseño responsivo y adaptable:

- UI adaptada a diferentes tamaños de pantalla con el uso de ConstraintLayout o LinearLayout,
- Interfaz acepta diferentes resoluciones y orientaciones (portrait/landscape),
- Dimensionalidad relativa (dp, sp) en lugar de valores fijos (px).

5. La experiencia de usuario (UX):

- Navegación intuitiva y fácil de usar,
- Paleta de colores y tipografía coherentes y legibles,
- Interfaz considera los principios de Material Design.

6. Las pruebas en emulador o dispositivo real:

- Emulador o dispositivo físico muestra la interfaz de la aplicación sin errores al renderizarla.

7. **Distribución de la aplicación móvil para Android o IOS**, que considere los siguientes elementos.

1. Aplicación preparada para su distribución:

- Archivos APK/AAB (Android) o IPA (iOS) generados,

- Número de versión y código de compilación establecido a partir de las normas de cada plataforma,
- Aplicación optimizada para mejorar el rendimiento de los recursos (reducción de tamaño, imágenes en WebP, etc.),
- Prueba de compatibilidad realizadas en diferentes dispositivos y versiones de Android/iOS,
- Pruebas de seguridad realizadas para evitar vulnerabilidades en la app.

2. Publicación de la app en Google Play Store:

- Cuenta de desarrollador en Google Play Console creada,
- Formulario de contenido y políticas de privacidad requisitado,
- Aplicación en formato AAB (Android App Bundle) o APK subida a la plataforma,
- Permisos de la aplicación y su revisión en Google Play configurada,
- Capturas de pantalla y videos promocionales de la app subidas a la plataforma,
- Descripción clara y concisa de la aplicación en Play Store,
- Prueba beta cerrada o abierta antes de la publicación oficial realizada,
- Aplicación verificada antes de la publicación considerando las normas de Google Play.

3. Publicación de la app en Apple App Store:

- Cuenta de desarrollador en Apple Developer Program creada,
- Perfil de aprovisionamiento y certificado de distribución generado,
- Aplicación en formato IPA subida al App Store Connect,
- Información de la aplicación (nombre, descripción, categoría, capturas de pantalla, etc.) capturada,
- Pruebas configuradas en TestFlight para versión beta antes de la publicación,
- Aplicación verificada a partir de las directrices de revisión de Apple.

4. Distribución personal de la app:

- App compartida desde espacios personales (Drive, GitHub, etc.) mediante enlaces directos,
- Archivo APK/IPA compartidos en medios alternativos (Telegram, WhatsApp, email, etc.),
- Aplicación descargada e instalada en dispositivos sin acceso a tiendas oficiales.

Comprobando mis aprendizajes

Rúbrica de Evaluación: Desarrollo de App de Hábitos Saludables (valor total 110pts)

1. Diagrama UML de Clases (15 puntos)

| CRITERIO DE EVALUACIÓN | DEFICIENTE (1 O 2 PTS) | REGULAR (3 PTS.) | BUENO (4 PTS.) | EXCELENTE (5 PTS.) | PUNTAJE |
|--------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|--|---------|
| Estructura de clases | Estructura incompleta o confusa | Faltan tres elementos | Faltan uno o dos elementos | Todas las clases tienen nombre, atributos y métodos bien definidos | |
| Relaciones entre clases | Relaciones ausentes o incorrectas | Relaciones confusas, sin claridad | Relaciones parcialmente correctas | Se representan correctamente herencia, asociación, agregación, composición | |
| Nomenclatura UML | No cumple con la notación | Notación confusa, sin claridad | Cumple parcialmente en tres casos | Cumple completamente con la notación estándar | |

2. Diagrama UML de Objetos (15 puntos)

| CRITERIO DE EVALUACIÓN | DEFICIENTE (1 O 2 PTS) | REGULAR (3 PTS.) | BUENO (4 PTS.) | EXCELENTE (5 PTS.) | PUNTAJE |
|---------------------------------|-----------------------------------|------------------------|--------------------------------|--|---------|
| Instancias y atributos | Objetos mal definidos | Valores incompletos | Faltan uno o dos valores | Objetos bien definidos con valores asignados | |
| Relaciones entre objetos | Relaciones incorrectas o ausentes | Relaciones poco claras | Una o dos relaciones correctas | Claras y coherentes con el diagrama de clases | |
| Notación UML | No cumple con la notación | Notación confusa | Cumple en dos o tres casos | Cumple con la notación estándar en todos los casos | |

3. Programas Básicos en Java (25 puntos)

| CRITERIO DE EVALUACIÓN | DEFICIENTE (1 O 2 PTS) | REGULAR (3 PTS.) | BUENO (4 PTS.) | EXCELENTE (5 PTS.) | PUNTAJE |
|--------------------------------------|-----------------------------|--------------------------|-------------------------|--|---------|
| Estructura del proyecto | Sin estructura clara | Organización deficiente | Organización parcial | IDE configurado correctamente con carpetas organizadas | |
| Código modular y reutilizable | Código desorganizado | Código poco reutilizable | Modularidad parcial | Clases y métodos bien estructurados | |
| Sintaxis y tipos de datos | Uso incorrecto generalizado | Errores frecuentes | Algunos errores menores | Uso correcto de variables, operadores y estructuras | |
| Manejo de excepciones | Sin manejo de excepciones | Manejo básico | Manejo parcial | Implementación efectiva de try-catch-finally | |
| Interacción y pruebas | Sin pruebas ni interacción | Interacción limitada | Interfaz básica | Interfaz funcional y pruebas realizadas | |

4. Documento de Planeación del Proyecto (20 puntos)

| CRITERIO DE EVALUACIÓN | DEFICIENTE (1 O 2 PTS) | REGULAR (3 PTS.) | BUENO (4 PTS.) | EXCELENTE (5 PTS.) | PUNTAJE |
|---|------------------------|---------------------------|------------------------|--------------------------------------|---------|
| Definición del problema y objetivos | Ausentes o confusos | Poco claros | Parcialmente claros | Claros, relevantes y bien redactados | |
| Análisis de mercado y apps similares | Sin análisis | Investigación superficial | Investigación parcial | Investigación completa y pertinente | |
| Requerimientos y factibilidad | Ausentes o incorrectos | Poco claros | Parcialmente definidos | Bien definidos y justificados | |
| Diagrama de Gantt o herramienta de gestión | Ausente o incorrecto | Poco claro | Parcialmente completo | Completo y bien estructurado | |

5. Wireframes y Mockups (15 puntos)

| CRITERIO DE EVALUACIÓN | DEFICIENTE (1 O 2 PTS) | REGULAR (3 PTS.) | BUENO (4 PTS.) | EXCELENTE (5 PTS.) | PUNTAJE |
|-------------------------------------|------------------------|-----------------------|--------------------------|--|---------|
| Wireframes de baja fidelidad | Ausentes o confusos | Poco claros | Estructura parcial | Claros, funcionales y bien estructurados | |
| Mockups de alta fidelidad | Diseño deficiente | Diseño básico | Consistencia parcial | Visualmente consistentes y detallados | |
| Prototipo interactivo y UX | Sin prototipo | Navegación deficiente | Flujo parcialmente claro | Flujo intuitivo, accesible y funcional | |

6. Interfaz Gráfica en Android Studio (10 puntos)

| CRITERIO DE EVALUACIÓN | DEFICIENTE (1 O 2 PTS) | REGULAR (3 PTS.) | BUENO (4 PTS.) | EXCELENTE (5 PTS.) | PUNTAJE |
|---------------------------------|----------------------------|---------------------|-----------------------|--|---------|
| Diseño UI y organización | Layouts desorganizados | Layouts poco claros | Organización parcial | Layouts bien estructurados y nombrados | |
| Diseño responsivo y UX | No responsivo ni funcional | Diseño básico | Adaptabilidad parcial | Adaptable, intuitivo y visualmente coherente | |

7. Distribución de la Aplicación (10 puntos)

| CRITERIO DE EVALUACIÓN | DEFICIENTE (1 O 2 PTS) | REGULAR (3 PTS.) | BUENO (4 PTS.) | EXCELENTE (5 PTS.) | PUNTAJE |
|------------------------------------|------------------------------------|----------------------|-----------------------|---|---------|
| Preparación técnica | Archivos incompletos o con errores | Archivos básicos | Optimización parcial | Archivos generados correctamente y optimizados | |
| Publicación y documentación | Sin publicación o documentación | Documentación básica | Parcialmente completo | Cumple con requisitos de Google/Apple Store o distribución personal | |

Bibliografía.

- Alarcón, J. (2021). *Conceptos fundamentales sobre programación orientada a objetos*. CampusMVP. <https://www.campusmvp.es/recursos/post/los-conceptos-fundamentales-sobre-programacion-orientada-objetos-explicados-de-manera-simple.aspx>
- Booch, G., Rumbaugh, J., y Jacobson, I. (2006). *El lenguaje unificado de modelado (UML)*. Pearson Educación.
- DataScience. (2024). *Estructuras de control en Java*. DataScience. <https://datascience.com/estructuras-de-control-en-java/>
- UNAM. (s.f.). *Arreglos unidimensionales y multidimensionales en Java*. Repositorio UAPA. <https://uapa.cuaed.unam.mx/sites/default/files/minisite/static/5b8d3d3f-ca5b-4afc-85df-d88d4619903e/UAPA-arreglos-unidimensionales-multidimensionales/index.html>
- _____. (s. f.). *Diagramas UML*. Repositorio UAPA. <https://uapa.cuaed.unam.mx/sites/default/files/minisite/static/2697a297-363e-4794-a4cc-04a2775031b6/UAPA-UML/index.html>
- _____. (s.f.). *Fundamentos y sintaxis de Java*. Repositorio UAPA. <https://uapa.cuaed.unam.mx/sites/default/files/minisite/static/5c89e6f5-c58b-4d70-99ed-90babe2ca1d4/UAPA-fundamentos-sintaxis-lenguaje/index.html>
- Carhuapoma, L. (2023). ¿Qué es un mockup? ImagineApps. <https://www.imagineapps.co/blog-posts-es/que-es-un-mockup>
- Coursera Staff. (2023). ¿Qué es un wireframe?, Como crear uno. Coursera. <https://www.coursera.org/mx/articles/wireframe>
- Custódio, F. (2023). Etapas del proceso de desarrollo de aplicaciones. FWtecnología. <https://fwtecnologia.com/es/blog/post/pasos-proceso-desarrollo-aplicaciones>
- García, J. (2020). Desarrollo de aplicaciones móviles en México: Tendencias y oportunidades. Editorial Tecnológica.
- Midulive. (16 de agosto de 2023). *¡Adictos Mockups GRATIS para TODOS!*. [Archivo de Video]. YouTube. <https://www.youtube.com/watch?v=KPFSw2I7TC4>
- Moure, B. (20 de enero de 2023). Cómo comenzar a diseñar una app. [Archivo de Video]. YouTube. <https://www.youtube.com/watch?v=4Azc4Gs2nIU>
- Purita, G. (2023). Experiencia de usuario: Qué es y por qué es importante. BusinessSchool. <https://www.obsbusiness.school/blog/experiencia-de-usuario-que-es-y-por-que-es-tan-importante>
- Tangramconsulting. (2024). Consejos para crear una app que resuelva problemas reales. Tangramconsulting. <https://tangramconsulting.es/noticias/consejos-practicos-para-crear-una-app-que-resuelva-problemas-reales>

Developers. (2024). Ciclo de vida de la actividad. AndroidDevelopers.
<https://developer.android.com/guide/components/activities/activity-lifecycle?hl=es-419>

Developers. (2024). Componentes principales de Android Studio. AndroidDevelopers.
<https://developer.android.com/develop/ui/views/components/button?hl=es-419>

Develou. (2023). *Aprendiendo sobre la Arquitectura Android*. Develou.
<https://www.develou.com/aprendiendo-la-arquitectura-de-android/>

EdumasterRD. (17 de enero de 2024). Crea apps adaptables a todos los dispositivos. [Archivo de Video]. YouTube. <https://www.youtube.com/watch?v=3LQbAd093dA>

Fernández, Y. (2025). Emuladores de Android. Xataka.
<https://www.xataka.com/basics/mejores-emuladores-android-para-pc>

García, A. (2022). Desarrollo de aplicaciones Android para el mercado mexicano. Alfaomega.

Moreno, H. (13 de agosto de 2024). Ejemplo con Android Studio y Java. [Archivo de Video]. YouTube. <https://www.youtube.com/watch?v=f8Bx2QhJ8d4>