

GUIA #1

# FUNDAMENTOS DE PROGRAMACIÓN AVANZADA

TALLER DE PROGRAMACIÓN  
OLGA LUCÍA ROA BOHÓRQUEZ

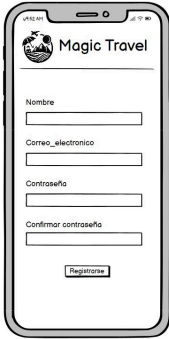
KAROL STEFANY ORDOÑEZ PEÑALOZA  
YEFERSON PIÑEROS PEÑA

10/08/2024


## INTRODUCCIÓN

Se requiere desarrollar un sistema de información para una agencia de turismo, para la cual se debe elaborar una tabla por cada requerimiento funcional, se debe especificar 10 requerimientos funcionales.


### REQUERIMIENTOS FUNCIONALES

IDENTIFICADOR:	NOMBRE:
RF01	Registrar usuarios
REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:	DOCUMENTOS DE VISUALIZACIÓN ASOCIADOS:
RF02	
ENTRADA:	SALIDA:
<ul style="list-style-type: none"><li>- nombre: String (30 caracteres)</li><li>- correo_electronico: String (20 caracteres)</li><li>- contraseña: String (8-20 caracteres, debe incluir al menos una mayúscula, un número y un carácter especial)</li></ul>	<ul style="list-style-type: none"><li>- Registro de usuario creado en la base de datos.</li><li>- Respuesta: String (“Registro exitoso” o mensaje de error).</li></ul>
DESCRIPCIÓN:	
<ul style="list-style-type: none"><li>- Este requerimiento permite a un nuevo usuario registrarse en el sistema proporcionando sus datos personales. El registro es esencial para que el usuario pueda acceder a otras funcionalidades del sistema.</li></ul>	

MANEJO DE SITUACIONES ANORMALES:	
<ul style="list-style-type: none"> <li>- Situación: El usuario intenta registrarse con un correo electrónico ya registrado.</li> <li>- Comportamiento del Sistema: Mostrar un mensaje de error "El correo electrónico ya está registrado." y no permitir la creación de la cuenta.</li> </ul>	
CRITERIOS DE ACEPTACIÓN:	
<ul style="list-style-type: none"> <li>- Para entradas válidas (nombre, correo, contraseña), el sistema debe crear un nuevo usuario y persistir los datos correctamente en la base de datos.</li> <li>- Para entradas inválidas (correo ya registrado, contraseña no cumple requisitos), el sistema debe mostrar mensajes de error específicos y no crear la cuenta.</li> <li>- Verificación de que el nuevo usuario puede iniciar sesión exitosamente, asegurando la persistencia de los datos.</li> </ul>	

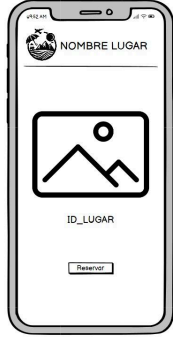
IDENTIFICADOR:	NOMBRE:
RF02	Iniciar sesión
REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:	DOCUMENTOS DE VISUALIZACIÓN ASOCIADOS:
	
ENTRADA:	SALIDA:
<ul style="list-style-type: none"> <li>- correo_electronico: String (20 caracteres)</li> <li>- contraseña: String (8-20 caracteres)</li> </ul>	<ul style="list-style-type: none"> <li>- Respuesta: String ("Inicio de sesión exitoso" o mensaje de error)</li> </ul>


DESCRIPCIÓN:	
<ul style="list-style-type: none"> <li>- Este requerimiento permite a un usuario registrado iniciar sesión en el sistema utilizando su correo electrónico y contraseña.</li> </ul>	
MANEJO DE SITUACIONES ANORMALES:	
<ul style="list-style-type: none"> <li>- Situación: El usuario ingresa un correo electrónico no registrado.</li> <li>- Comportamiento del Sistema: Mostrar un mensaje de error "Correo electrónico no registrado." y no permitir el inicio de sesión.</li> <li>- Situación: El usuario ingresa una contraseña incorrecta.</li> <li>- Comportamiento del Sistema: Mostrar un mensaje de error "Contraseña incorrecta." y no permitir el inicio de sesión.</li> </ul>	
CRITERIOS DE ACEPTACIÓN:	
<ul style="list-style-type: none"> <li>- Para entradas válidas (correo registrado, contraseña correcta), el sistema debe iniciar la sesión.</li> <li>- Para entradas inválidas (correo no registrado, contraseña incorrecta), el sistema debe mostrar mensajes de error y no iniciar sesión.</li> </ul>	

IDENTIFICADOR:	NOMBRE:
RF03	Buscar lugares turísticos
REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:	DOCUMENTOS DE VISUALIZACIÓN ASOCIADOS:
RF04	

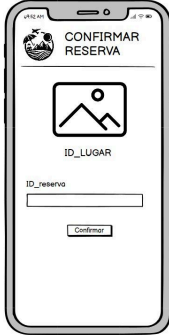
ENTRADA:	SALIDA:
<ul style="list-style-type: none"> <li>- ubicación: String (30 caracteres)</li> <li>- tipo: String (16 caracteres)</li> <li>- rango_precio: String ("min-max")</li> <li>- disponibilidad: Boolean (true/false)</li> </ul>	<ul style="list-style-type: none"> <li>- Lista de lugares turísticos que cumplen con los criterios de búsqueda</li> </ul>
DESCRIPCIÓN:	
<ul style="list-style-type: none"> <li>- Este requerimiento permite a los usuarios buscar lugares turísticos en la base de datos utilizando diferentes filtros como ubicación, tipo de atracción, rango de precio y disponibilidad.</li> </ul>	
MANEJO DE SITUACIONES ANORMALES:	
<ul style="list-style-type: none"> <li>- Situación: El usuario no proporciona criterios válidos para la búsqueda (ej. valores de rango de precio incorrectos).</li> <li>- Comportamiento del Sistema: Mostrar un mensaje de error "Por favor, ingrese criterios de búsqueda válidos." y no ejecutar la búsqueda.</li> </ul>	
CRITERIOS DE ACEPTACIÓN:	
<ul style="list-style-type: none"> <li>- Para entradas válidas (filtros de búsqueda correctos), el sistema debe mostrar una lista de lugares que cumplen con los criterios.</li> <li>- Para entradas inválidas (filtros incorrectos), el sistema debe mostrar mensajes de error específicos y no realizar la búsqueda.</li> </ul>	

IDENTIFICADOR:	NOMBRE:
RF04	Visualizar detalles de lugares turísticos
REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:	DOCUMENTOS DE VISUALIZACIÓN ASOCIADOS:

	
ENTRADA:	SALIDA:
<ul style="list-style-type: none"> <li>- id_lugar: int (20 entero positivo)</li> </ul>	<ul style="list-style-type: none"> <li>- Detalles completos del lugar turísticos</li> </ul>
DESCRIPCIÓN:	
<ul style="list-style-type: none"> <li>- Este requerimiento permite al usuario ver información detallada de un lugar turístico específico, como descripción, imágenes, y disponibilidad.</li> </ul>	
MANEJO DE SITUACIONES ANORMALES:	
<ul style="list-style-type: none"> <li>- Situación: El usuario intenta acceder a los detalles de un lugar turístico inexistente (ID incorrecto).</li> <li>- Comportamiento del Sistema: Mostrar un mensaje de error "Lugar turístico no encontrado." y no cargar los detalles.</li> </ul>	
CRITERIOS DE ACEPTACIÓN:	
<ul style="list-style-type: none"> <li>- Para entradas válidas (ID de lugar válido), el sistema debe mostrar toda la información detallada del lugar turístico.</li> <li>- Para entradas inválidas (ID de lugar no válido), el sistema debe mostrar un mensaje de error y no cargar los detalles.</li> <li>- Verificación de que toda la información mostrada corresponde con los datos en la base de datos.</li> </ul>	
IDENTIFICADOR:	NOMBRE:

RF05	Reservar servicios turísticos
REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:	DOCUMENTOS DE VISUALIZACIÓN ASOCIADOS:
RF04	
ENTRADA:	SALIDA:
<ul style="list-style-type: none"> <li>- id_lugar(int positivo)</li> <li>- fecha_inicio(formato YYYY-MM-DD)</li> <li>- fecha_fin(formato YYYY-MM-DD)</li> <li>- num_personas(int 1-20)</li> </ul>	<ul style="list-style-type: none"> <li>- Creación de una reserva en la base de datos</li> <li>- Respuesta: String (“Reserva exitosa” o mensaje de error)</li> </ul>
DESCRIPCIÓN:	
<ul style="list-style-type: none"> <li>- Este requerimiento permite a los usuarios reservar un servicio turístico para unas fechas y número de personas específicos.</li> </ul>	
MANEJO DE SITUACIONES ANORMALES:	
<ul style="list-style-type: none"> <li>- Situación: El usuario intenta reservar un lugar que no está disponible en las fechas seleccionadas.</li> <li>- Comportamiento del Sistema: Mostrar un mensaje de error "El lugar no está disponible para las fechas seleccionadas." y no proceder con la reserva.</li> </ul>	
CRITERIOS DE ACEPTACIÓN:	
<ul style="list-style-type: none"> <li>- Para entradas válidas (fechas, número de personas, ID de lugar válido y disponible), el sistema debe crear una reserva y persistir correctamente en la base de datos.</li> <li>- Para entradas inválidas (fechas no disponibles, número de personas fuera del rango), el sistema debe mostrar mensajes de error y no crear la reserva.</li> </ul>	

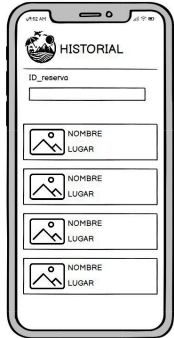
<ul style="list-style-type: none"> <li>- Verificación de que la reserva creada refleja correctamente la información ingresada y se guarda de manera persistente.</li> <li>- Verificación de que el sistema calcula y muestra correctamente el costo total de la reserva.</li> </ul>
---

IDENTIFICADOR:	NOMBRE:
RF06	Confirmar reservas
REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:	DOCUMENTOS DE VISUALIZACIÓN ASOCIADOS:
	
ENTRADA:	SALIDA:
<ul style="list-style-type: none"> <li>- id_reserva(int positivo)</li> </ul>	<ul style="list-style-type: none"> <li>- Verificación de que la reserva ha sido creada y confirmada</li> <li>- Respuesta: String ("Confirmación exitosa" o mensaje error)</li> </ul>
DESCRIPCIÓN:	
<ul style="list-style-type: none"> <li>- Este requerimiento asegura que una reserva se ha creado y confirmado correctamente en la base de datos.</li> </ul>	
MANEJO DE SITUACIONES ANORMALES:	
<ul style="list-style-type: none"> <li>- Situación: El usuario intenta confirmar una reserva inexistente (ID incorrecto).</li> <li>- Comportamiento del Sistema: Mostrar un mensaje de error "Reserva no encontrada." y no proceder con la confirmación.</li> </ul>	

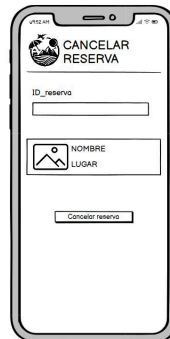


### CRITERIOS DE ACEPTACIÓN:


- Para entradas válidas (ID de reserva válida y pendiente), el sistema debe confirmar la reserva y actualizar su estado en la base de datos.
- Para entradas inválidas (ID de reserva no válida o ya confirmada), el sistema debe mostrar un mensaje de error y no confirmar la reserva.
- Verificación de que el estado de la reserva se actualiza correctamente a "Confirmada" en la base de datos.

IDENTIFICADOR:	NOMBRE:
RF07	Visualizar historial de reservas
REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:	DOCUMENTOS DE VISUALIZACIÓN ASOCIADOS:
RF05	
ENTRADA:	SALIDA:
<ul style="list-style-type: none"> <li>- id_reserva(int positivo)</li> </ul>	<ul style="list-style-type: none"> <li>- Lista de reservas del usuario</li> </ul>
DESCRIPCIÓN:	
<ul style="list-style-type: none"> <li>- Permite a los usuarios ver un historial de todas las reservas que han realizado, con detalles como las fechas y el estado de la reserva.</li> </ul>	
MANEJO DE SITUACIONES ANORMALES:	

<ul style="list-style-type: none"> <li>- Situación: El usuario intenta visualizar un historial inexistente (aún no ha reservado).</li> <li>- Comportamiento del Sistema: Mostrar un mensaje de error "Aún no hay reservas."</li> </ul>
CRITERIOS DE ACEPTACIÓN:
<ul style="list-style-type: none"> <li>- Para entradas válidas (ID de reserva), el sistema debe mostrar todo el historial de reservas del usuario de manera precisa.</li> <li>- Para entradas inválidas (ID de reserva no válido o acceso no autorizado), el sistema debe mostrar un mensaje de error y no mostrar el historial.</li> <li>- Verificación de que todos los datos de reservas se presentan de manera consistente con la base de datos.</li> </ul>

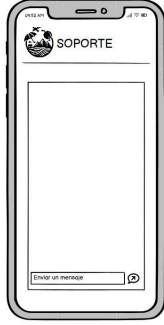
IDENTIFICADOR:	NOMBRE:
RF08	Cancelar reservas
REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:	DOCUMENTOS DE VISUALIZACIÓN ASOCIADOS:
RF05	
ENTRADA:	SALIDA:
<ul style="list-style-type: none"> <li>- id_reserva(int positivo)</li> </ul>	<ul style="list-style-type: none"> <li>- Actualización del estado de la reserva en la base de datos (estado: "Cancelada" )</li> <li>- Respuesta: String ("Reserva cancelada" o mensaje error)</li> </ul>
DESCRIPCIÓN:	

<ul style="list-style-type: none"> <li>- Este requerimiento permite a los usuarios cancelar una reserva que hayan realizado, dentro de los límites permitidos por las políticas del sistema.</li> </ul>
MANEJO DE SITUACIONES ANORMALES:
<ul style="list-style-type: none"> <li>- Situación: El usuario intenta cancelar una reserva fuera del plazo permitido.</li> <li>- Comportamiento del Sistema: Mostrar un mensaje de error "No se puede cancelar la reserva fuera del plazo permitido." y no proceder con la cancelación.</li> </ul>
CRITERIOS DE ACEPTACIÓN:
<ul style="list-style-type: none"> <li>- Para entradas válidas (ID de reserva válida y dentro del plazo de cancelación), el sistema debe cancelar la reserva y actualizar su estado en la base de datos.</li> <li>- Para entradas inválidas (ID de reserva no válida o fuera del plazo permitido), el sistema debe mostrar un mensaje de error y no cancelar la reserva.</li> <li>- Verificación de que la reserva se actualiza correctamente a "Cancelada" y que cualquier penalización se calcula y se aplica de manera adecuada.</li> </ul>

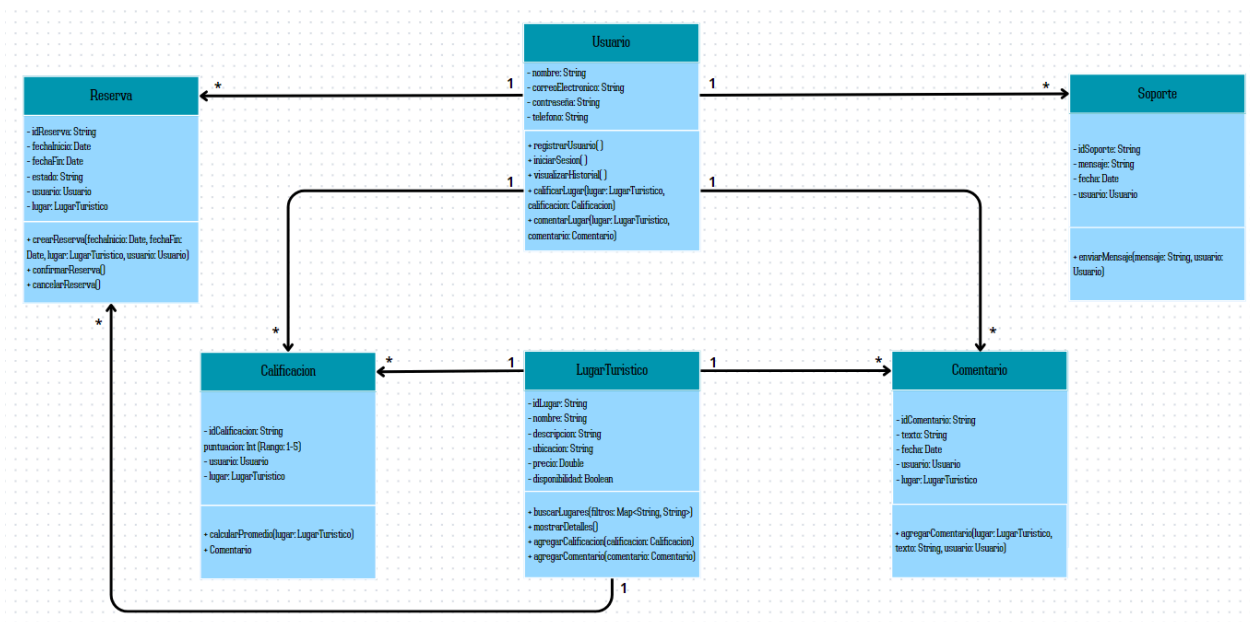
IDENTIFICADOR:	NOMBRE:
RF09	Calificar y comentar lugares turísticos
REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:	DOCUMENTOS DE VISUALIZACIÓN ASOCIADOS:
	
ENTRADA:	SALIDA:
<ul style="list-style-type: none"> <li>- id_lugar(int positivo)</li> <li>- calificación(int positivo (1-5))</li> </ul>	<ul style="list-style-type: none"> <li>- Registro de la calificación y comentario en la base de datos</li> </ul>

- comentario(String 100 (máximo opcional) )	- Respuesta: String ("Calificación y comentario registrados" o mensaje error)
DESCRIPCIÓN:	
- Permite a los usuarios dejar una calificación y un comentario sobre un lugar turístico que han visitado.	
MANEJO DE SITUACIONES ANORMALES:	
<ul style="list-style-type: none"> <li>- Situación: El usuario intenta calificar un lugar que no ha visitado (no tiene una reserva en dicho lugar).</li> <li>- Comportamiento del Sistema: Mostrar un mensaje de error "No puede calificar un lugar que no ha visitado." y no permitir la calificación.</li> </ul>	
CRITERIOS DE ACEPTACIÓN:	
<ul style="list-style-type: none"> <li>- Para entradas válidas (ID de lugar válido, calificación y comentario dentro del rango permitido), el sistema debe guardar la calificación y comentario en la base de datos.</li> <li>- Para entradas inválidas (calificación fuera de rango o comentario demasiado largo), el sistema debe mostrar un mensaje de error y no guardar la calificación/comentario.</li> <li>- Verificación de que la nueva calificación promedio del lugar se recalcula y se actualiza correctamente en la base de datos.</li> <li>- Verificación de que los comentarios y calificaciones se guardan de manera persistente y segura.</li> </ul>	

IDENTIFICADOR:	NOMBRE:
RF10	Ofrecer soporte en línea
REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:	DOCUMENTOS DE VISUALIZACIÓN ASOCIADOS:

	
ENTRADA:	SALIDA:
<ul style="list-style-type: none"> <li>- mensaje: (String 100 (máximo opcional))</li> </ul>	<ul style="list-style-type: none"> <li>- Envío del mensaje a soporte</li> <li>- Respuesta: String ("Mensaje enviado" o mensaje error)</li> </ul>
DESCRIPCIÓN:	
<ul style="list-style-type: none"> <li>- Este requerimiento permite a los usuarios enviar mensajes al equipo de soporte para obtener ayuda o resolver problemas.</li> </ul>	
MANEJO DE SITUACIONES ANORMALES:	
<ul style="list-style-type: none"> <li>- Situación: El usuario intenta enviar un mensaje de soporte con campos obligatorios vacíos.</li> <li>- Comportamiento del Sistema: Resaltar los campos vacíos y mostrar un mensaje "Este campo es obligatorio." No permitir el envío del mensaje.</li> </ul>	
CRITERIOS DE ACEPTACIÓN:	
<ul style="list-style-type: none"> <li>- Para entradas válidas (mensaje dentro de los límites permitidos), el sistema debe crear y guardar el mensaje.</li> <li>- Para entradas inválidas (campos vacíos o mensaje inapropiado), el sistema debe mostrar un mensaje de error y no enviar el mensaje de soporte.</li> </ul>	

Se requiere Diseñar el sistema de gestión de libros a través de un diagrama de clases aplicando UML (Lenguaje de Modelado Unificado)



## PREGUNTAS ORIENTADORAS

¿Cuáles fueron los aprendizajes obtenidos al realizar esta guía?, liste como mínimo 3 aprendizajes y relaciónelos con su futuro quehacer profesional.

- La definición precisa de requerimientos funcionales es crucial para el éxito de un proyecto de software, ya que evita confusiones y asegura que el desarrollo satisfaga las expectativas del cliente.
- Planificar cómo manejar errores y situaciones imprevistas es clave para crear sistemas robustos y confiables.
- Establecer criterios de aceptación claros permite verificar que los requerimientos se cumplen adecuadamente y que el sistema funciona como se espera.

¿Dónde presentó mayor dificultad resolviendo la guía? y ¿cómo lo resolvieron? ¿Cuáles fueron las estrategias de solución?

- Se presentó dificultad en la elaboración del video en inglés ya que el uso de un lenguaje técnico en otro idioma es más complejo que el tradicional. La solución a este problema fue reescribir varias veces el diálogo, escucharlo y repetirlo para posteriormente grabarlo en fragmentos.

- Realizando algunos requerimientos. La solución fue pensar en los procesos desde un punto de vista diferente al del cliente y apoyarnos de la idea de un proceso que requiere un paso a paso y diversas condiciones para que se cumpla un cometido.

## ACTIVIDAD DE TRABAJO AUTÓNOMO

Realice la lectura del siguiente artículo: Keepcoding Tech School. (2024). ¿Qué son los paradigmas de programación y qué tipos hay?

<https://keepcoding.io/blog/paradigmas-de-programacion/> (Recurso de Internet)

Enuncie y explique brevemente los paradigmas de programación propuestos en dicho artículo, investigando la información solicitada en el siguiente cuadro comparativo.

Paradigma de Programación.	Ventajas.	Desventajas.	Lenguaje de programación.
Programación Imperativa.	1. Es fácil de entender ya que sigue una secuencia de instrucciones claras. 2. Permite un control preciso en la ejecución y manipulación de las cosas. 3. Es más eficiente en rendimiento.	1. Se llega a complicar a medida que el programa crece. 2. Los errores suelen ser difíciles de rastrear. 3. No siempre se puede implementar la recursividad.	C, Pascal.

Paradigma Declarativa.	<ol style="list-style-type: none"> <li>1. Se puede ir directo al objetivo sin necesidad de describir el proceso.</li> <li>2. Reduce los errores de manejo del estado (variables que se modifican en el proceso).</li> <li>3. Los compiladores pueden trabajar de manera óptima.</li> </ol>	<ol style="list-style-type: none"> <li>1. No es intuitivo para su aprendizaje.</li> <li>2. Puede ser menos eficientes al hacer uso de abstracciones</li> <li>3. suele ser usado para problemas en específico.</li> </ol>	Haskel para programación funcional y Prolog para programación lógica.
Programación Orientada a Objetos (OOP).	<ol style="list-style-type: none"> <li>1. Facilita la organización del código y a su vez la recursividad.</li> <li>2. Pilares como herencia y polimorfismo permiten flexibilidad y reutilización.</li> <li>3. Puede mantenerse el código.</li> </ol>	<ol style="list-style-type: none"> <li>1. Puede hacerse muy complejo para proyectos simples.</li> <li>2. Pilares como la abstracción pueden afectar el rendimiento en la compilación.</li> <li>3. Un diseño mal hecho o implementado puede afectar a futuro el mantenimiento.</li> </ol>	Python, Java, C++.



Programación Orientada a Aspectos (AOP).	<ol style="list-style-type: none"> <li>1. Maneja de mejor manera aspectos como la seguridad y los registros al separarlos.</li> <li>2. Los aspectos pueden usarse de manera recursiva.</li> <li>3. Fácil de mantener y actualizar ya que los cambios tienden a realizarse en un mismo lugar donde se agrupan las actividades de gestión.</li> </ol>	<ol style="list-style-type: none"> <li>1. Incrementa la complejidad del código y su entendimiento.</li> <li>2. Tiene muy pocas herramientas en comparación con los otros paradigmas.</li> <li>3. puede sobrecargarse en la ejecución y ser lento.</li> </ol>	AspectJ.
Programación Funcional.	<ol style="list-style-type: none"> <li>1. Facilita el razonamiento del código.</li> <li>2. Sus funciones pueden ser editadas de manera sencilla.</li> <li>3. Maneja variables inmutables (no cambian en el tiempo) lo que facilita la ejecución de tareas en paralelo.</li> </ol>	<ol style="list-style-type: none"> <li>1. Es uno de los paradigmas más difíciles de aprender en especial para quienes manejan “programación imperativa”</li> <li>2. A su vez, la inmutabilidad afecta el rendimiento.</li> <li>3. Tiene pocas herramientas y menos bibliotecas.</li> </ol>	Lisp, Haskell, Erlang.

Programación Lógica.	<ol style="list-style-type: none"> <li>1. Especifica lo que se desea sin detallar en el proceso.</li> <li>2. Es fácil solucionar cualquier tipo de errores que se produzcan sobre la marcha.</li> <li>3. Maneja bases sólidas y simples.</li> </ol>	<ol style="list-style-type: none"> <li>1. No es muy eficiente a comparación de otros paradigmas.</li> <li>2. Tiene una curva de aprendizaje baja.</li> <li>3. Tiende a solucionar problemas muy específicos.</li> </ol>	Prolog.
Programación Reactiva.	<ol style="list-style-type: none"> <li>1. Responde a eventos en tiempo real, lo que hace que reaccione rápido a cambios y actualizaciones inmediatas.</li> <li>2. Maneja de manera adecuada la concurrencia lo que le permite responder bien a tareas simultáneas.</li> <li>3. Facilita el mantenimiento y mantiene un flujo de datos claros.</li> </ol>	<ol style="list-style-type: none"> <li>1. Suele ser difícil de entender y a su vez difícil de depurar.</li> <li>2. La gestión de todos los eventos puede ser costosa.</li> <li>3. A comparación de otros paradigmas este es el de menos soporte.</li> </ol>	Java (RxJava, ReactiveX Bibliotecas).

Programación Basada en Eventos.	1. El mejor paradigma en cuanto a respuestas de eventos. 2. tiene una gran escalabilidad para los usuarios. 3. En la programación basada en eventos, sus componentes pueden funcionar de manera independiente.	1. Es complejo de mantener. 2. Al no tener un flujo lineal es más difícil de depurar. 3. el mal manejo de este paradigma conduce directamente a problemas de rendimiento.	<sup>1</sup> “JavaScript en el contexto de desarrollo web es un ejemplo común de programación basada en eventos.”
---------------------------------	--	---	---

Los paradigmas son formas diferentes de realizar códigos para resolver determinados problemas. Cada paradigma tiene su manera de organizar y manejar la información, lo que afecta la estructura de un código y la manera en que enfrenta el problema. Como lo vimos anteriormente cuentan con características únicas lo que hace que enfrentar un problema pueda ser más fácil con uno u otro.

De manera popular, hacemos uso del paradigma de “Programación Orientada a Objetos (POO)” ya que usa muchos lenguajes de programación modernos, maneja de manera ordenada el código, es flexible y facilita el manejo de proyectos grandes.

Aun así, el paradigma más eficiente es el de “Programación Imperativa” ya que permite un control preciso sobre la manipulación del estado y flujo de ejecución. Lo anterior permite optimizar y usar mejor los recursos del sistema.

Ya por último tenemos al paradigma menos eficiente que es el de “Programación Lógica” y esto se debe a su constante sobrecarga y la necesidad de soluciones mediante retrocesos automáticos, además, es, computacionalmente, costoso y no es muy rápido resolviendo problemas.

## ACTIVIDAD DE COMPROBACIÓN DE TRABAJO AUTÓNOMO

The student makes a video IN ENGLISH of a minimum of 1 and a maximum of 3 minutes exposing a summary of the main idea of the theoretical framework, the consultation of the independent work activity and the code developed in the practical example proposed in the introductory process session 1 and 2. The scholar publishes the video in the activity FORUM.

### **DIALOGO DEL VIDEO:**

"hi everyone! Today, I'mma tell you abaout programming paradigms. But what exactly is a paradigm? Simply put, paradigms are different ways of writing code to solve problems. Each paradigm organizes and manages information uniquely, influencing the structure of your code and how it tackles specific challenges.

One of the most popular paradigms is Object-Oriented Programming, or OOP. OOP is used because it's modern, organizes code neatly, and is flexible, making it perfect for managing large projects. It revolves around creating objects that represent real-world entities, allowing for reusable code.

On the other hand, the Imperative Programming paradigm is considered the most efficient. It gives you precise control over how the program's state changes and how the execution flows. This control allows for better optimization and more efficient use of system resources, making it ideal for performance-critical applications.

Finally, we have Logic Programming, which is the least efficient paradigm. While it excels in solving complex problems with automatic backtracking, it comes with a significant computational cost and isn't as fast as the other paradigms. This makes it less practical for resource-intensive tasks.

In the end, choosing the right paradigm depends on the problem you're trying to solve. Each has its strengths and weaknesses, and understanding them can make your coding experience much smoother. Well, so it is all

Thanks for watching!"

**Link Video:**  Guia1 TDP.mp4

**"<https://drive.google.com/file/d/1QywAGi8q9fzSO5H4mdef5mUdk4vKAFQz/view?usp=sharing>"**

## **BIBLIOGRAFÍA**

1. <https://keepcoding.io/blog/paradigmas-de-programacion/> - Keepcoding Tech School. (2024).  
¿Qué son los paradigmas de programación y qué tipos hay?