

**Escala de apreciación: Evaluación Sumativa N°X**  
**Asignatura**  
**Unidad de aprendizaje**  
**(ponderación de evaluación XX%)**

Área Académica	Tecnologías de Información y Ciberseguridad		Carrera	Ingeniería y Análisis
Sede	Puente Alto		Código	TI2042
Docente	Miguel Acuña	Fecha	25/09/2024	
Sección	V-B50-N4-P12-C1/V	Duración	90	

Nombre Estudiante:				
		Apellido Paterno	Apellido Materno	Nombres
Rut:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Puntaje Máximo		Nota:	Firma Conforme	
Puntaje Obtenido				
Solicita Re-Corrección	Sí	No	Motivo:	

**INSTRUCCIONES GENERALES:**

1. La nota 4.0 se obtiene logrando un 60% del puntaje total.
2. Utilice lápiz pasta en sus respuestas.
3. Preocúpese de la redacción, ortografía y legibilidad de sus respuestas.

**Aprendizaje esperado:**

- 1.- Fundamentos de Programación, Lenguaje Dart: Aprende los conceptos básicos de Dart, el lenguaje de programación utilizado por Flutter.

**Criterios de evaluación:**

- 1.1.1.- Utiliza variables y operaciones del lenguaje según requerimiento, analizando situaciones problemáticas establecidas
- 1.1.2.- Codifica instrucciones, estructuras de decisión y operadores según requerimiento
- 1.1.3.- Codifica instrucciones utilizando fundamentos del lenguaje y aplicando sintaxis.
- 1.1.4.- Implementa una aplicación sencilla en código, según requerimiento.

**I. Presentación de la actividad**

Adquirir una comprensión sólida de los conceptos básicos de programación utilizando el lenguaje Dart, que es fundamental para el desarrollo con Flutter.

**II. Actividades**

La actividad consta de codificar en Dart a través de Visual Code en donde se le presentara un problema real a través de un enunciado del cual deberán interpretar y codificar según lo solicitado, aplicando conceptos básicos de dart.

**Actividad N°1:** La actividad consiste en que a través de un enunciado se le presentara un problema de la vida real y los estudiantes deberán codificar en Dart, aplicando conceptos básicos de dart como: (Uso y manejo de variable, ciclos de condición if-else), bucles, estructuras de datos (listas, array, map), funciones., la cual el estudiante deberá realizar la codificación a través de Visual Code, en donde al finalizar su ejecución deberá entregarlo en formato de texto, con su correspondiente identificación para ser revisado posteriormente.

**Actividad N°2:**

**Actividad N°3:**

**III. Requerimiento de presentación y entrega**

En este ítem de debe señalar la estructura del trabajo y/o presentación como, por ejemplo: uso de plantilla, formato de entrega, fecha de entrega, ortografía y redacción, medio de entrega, entre otros.

**Escala de apreciación: calidad.**

Criterio	Dimensión /Indicadores	Niveles de desempeño.				Observaciones
		Excelente (6 puntos)	Bueno (4 puntos)	Insatisfactorio (1 puntos)	Puntaje obtenido	
Referencia: Criterios de evaluación de descriptor de asignatura.	Dimensiones o indicadores derivados del descriptor de asignatura.					
Utiliza variables y operaciones del lenguaje según requerimiento, analizando situaciones problemáticas establecidas	Fundamentos de Programación Lenguaje Dart: Aprende los conceptos básicos de Dart, el lenguaje de programación utilizado por Flutter					
Codifica instrucciones, estructuras de decisión y operadores según requerimiento	Variables y tipos de datos O Estructuras de control (if, for, while					
Codifica instrucciones utilizando fundamentos del lenguaje y aplicando sintaxis	Funciones y métodos O Clases y objetos					
Implementa una aplicación sencilla en código, según requerimiento	Es capaz de implementar una solución requerida a través de requerimientos solicitados					

	<b>Puntaje total</b>					

**Escala de apreciación: Frecuencia.**

Criterio	Dimensión /Indicadores	Niveles de desempeño.				Observaciones
		Siempre (6 puntos)	A veces (4 puntos)	Nunca (1 puntos)	Puntaje obtenido	
Referencia: Criterios de evaluación de descriptor de asignatura.	Dimensiones o indicadores derivados del descriptor de asignatura.					

	<b>Puntaje obtenido</b>					

**Escala de apreciación: Intensidad.**

Criterio	Dimensión /Indicadores	Niveles de desempeño.				Observaciones
		Fuerte (6 puntos)	Moderado (4 puntos)	Débil (1 puntos)	Puntaje obtenido	
Referencia: Criterios de evaluación de descriptor de asignatura.	Dimensiones o indicadores derivados del descriptor de asignatura.					

	<b>Puntaje obtenido</b>					

### Tabla de puntaje y nota:

Pueden usar este formato o capturar la escala del siguiente link:

<https://escaladenotas.cl/>

Puntaje	Nota	Puntaje	Nota	Puntaje	Nota
0.0	1.0	10.0	3.1	20.0	5.8
1.0	1.2	11.0	3.3	21.0	6.1
2.0	1.4	12.0	3.5	22.0	6.4
3.0	1.6	13.0	3.7	23.0	6.7
4.0	1.8	14.0	3.9	24.0	7.0
5.0	2.0	15.0	4.2		
6.0	2.3	16.0	4.5		
7.0	2.5	17.0	4.8		
8.0	2.7	18.0	5.1		
9.0	2.9	19.0	5.4		

### Justificación de porque se cambió el programa

Aspecto	Flutter (Híbrido)	Desarrollo Nativo (Android)
Lenguaje de Programación	Dart	Java/Kotlin
Base de Código	Una sola base de código para iOS y Android	Bases de código separadas para iOS (Swift/Objective-C) y Android (Java/Kotlin)
Rendimiento	Alto rendimiento gracias a la compilación nativa y el motor de renderizado Skia	Rendimiento óptimo al estar completamente integrado con el sistema operativo
Curva de Aprendizaje	Relativamente rápida, especialmente si ya conoces Dart o tienes experiencia en desarrollo web	Puede ser más empinada debido a la necesidad de aprender las APIs específicas de Android
Desarrollo de UI	Amplia gama de widgets personalizables y consistentes en todas las plataformas	Uso de componentes nativos específicos de Android, lo que puede ofrecer una experiencia más auténtica
Tiempo de Desarrollo	Más rápido debido a la reutilización del código y el hot reload	Puede ser más lento debido a la necesidad de desarrollar y mantener dos bases de código separadas
Comunidad y Soporte	Gran comunidad y soporte activo de Google	Amplia comunidad y recursos disponibles, pero fragmentados entre diferentes lenguajes y plataformas
Acceso a Funcionalidades Nativas	Acceso a la mayoría de las funcionalidades nativas a través de plugins y paquetes	Acceso completo a todas las funcionalidades nativas sin necesidad de plugins adicionales
Actualizaciones y Mantenimiento	Más sencillo debido a la base de código única	Puede ser más complejo y costoso debido a la necesidad de mantener dos bases de código separadas