

(20 p.) Ud. se ha presentado a una convocatoria internacional de la NASA y accedió a una plaza como Cross-Platform Software Developer. El primer proyecto que le asignan es desarrollar una app en Flutter denominada **MyNasalInsights**, la cual debe cumplir con las siguientes especificaciones:

a. Uso de APIs según el código del estudiante:

- Si el último dígito del código de alumno es impar, deberán consumir datos del API APOD:
https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY&start_date=2017-07-08&end_date=2017-12-10.
-
- Si el último dígito del código de alumno es par, deberán consumir datos del API Mars Rover Photos:
https://api.nasa.gov/marsphotos/api/v1/rovers/curiosity/photos?sol=1000&api_key=DEMO_KEY.
- Genere su PROPIO API-KEY y revise la información aquí: <https://api.nasa.gov/>

b. Pantallas solicitadas:

Pantalla 1:

- Mostrar una imagen relacionada con el espacio como portada de la aplicación.
- Mostrar su nombre, foto, código de alumno y el nombre del API que le corresponde usar (APOD o Mars Rover Photos, dependiendo del último dígito de su código).
- Ojo: El nombre del API debe mostrarse mediante la implementación de un algoritmo, que extraiga el último dígito del código y según el valor muestre en un mensaje el nombre del API que corresponda.
- La pantalla 1 debe tener las secciones “Mostrar” y “Favoritos”. Estas secciones las puede implementar con botones o con un BottomNavigationBar.

Pantalla 2 - Mostrar:

- Presentar tres (3) atributos relevantes obtenidos del JSON del API correspondiente.
- Mostrar la imagen asociada al elemento consultado.
- Agregar la funcionalidad para guardar elementos en una lista de favoritos mediante un ícono o botón. El elemento se almacena en una base de datos local, utilizando sqflite, drift u otro ORM.

Pantalla 3 - Favoritos:

- Mostrar los elementos guardados como favoritos.
- Presentar dos (2) atributos relevantes diferentes a los presentados en la pantalla 2 y mostrar la imagen asociada al elemento consultado.
- Incluir la funcionalidad para eliminar elementos de la lista de favoritos mediante un ícono o botón (Puede o no tener la acción de swipe).

c. Requisitos técnicos:

- La aplicación debe contar con un ícono distintivo.
- Se tomará en cuenta (ver rúbrica de calificación adjunta):
 - el estado ejecutable de la aplicación, ○ el cumplimiento de las características solicitadas, ○ la aplicación de estándares de nomenclatura en inglés, o la interfaz de usuario,
 - la aplicación de los principios de diseño,
 - claridad y eficiencia de la lógica implementada, selección y manejo adecuado de carpetas, bibliotecas y paquetes, así como la organización aplicada para la codificación de clases.

Rúbrica

Criterio de Calificación	Excelente	Promedio	Deficiente
C01. Building y ejecución	Al abrir el proyecto y ordenar la ejecución, ésta se inicia sin problemas.	La aplicación no llega a desplegarse y ejecutarse en el emulador de Android Studio, sin embargo, el proceso de building llega a concluir.	Al cargar el proyecto el proceso de building presenta errores y no llega a concluir.
	2.0 puntos	1.0 punto	0 puntos
C02. UI & Pantalla 1	El proyecto evidencia en toda la experiencia móvil el uso de componentes compatibles y la aplicación de los principios fundamentales de Material Design. La vista "Pantalla 1" presenta una imagen de branding, su nombre, foto, código de alumno y el nombre del API que le corresponde usar (APOD o Mars Rover Photos, y los botones "Mostrar" y "Favoritos".	En el proyecto no se aplica en todas las vistas los principios de Material Design. La vista "Pantalla 1" está parcialmente diseñada.	No se percibe en ninguna parte de la experiencia móvil la aplicación de principios y componentes de Material Design. La Pantalla 1 no se encuentra diseñada.
	2.0 puntos	1.0 puntos	0 puntos
C03. Pantalla 2 / Networking	La aplicación presenta en la vista "Mostrar" el listado de todos los elementos y muestra un ícono que permite registrar el elemento como favorito en una base de datos SQLite.	La aplicación implementa Networking, sin embargo, no cumple completamente con las características solicitadas.	La aplicación no implementa Networking o el código no es funcional.
	5.0 puntos	2.5 punto	0 puntos
C04. Pantalla 3 / Registro de Favoritos	La vista "Favoritos" permite la visualización de los elementos que hayan sido registrados como favoritos, los que están en una base de datos local en SQLite en el dispositivo. Puede utilizar como framework de persistencia sqflite, drift u otro.	La aplicación implementa persistencia, siendo ésta funcional, pero no cumple completamente con las características solicitadas.	La aplicación no implementa persistencia.
	3.0 puntos	1.5 puntos	0 puntos
C05. Pantalla 3 / Borrado de Favoritos	La vista "Favoritos" muestra un ícono que permite eliminar el elemento registrado en la base de datos. La eliminación se realiza de manera correcta. (Puede o no tener la acción de swipe).	La vista muestra un ícono que permite eliminar el producto registrado en la base de datos. La aplicación implementa la eliminación, pero no se realiza de manera correcta.	La vista no cumple con lo solicitado.
	3.0 puntos	1.5 puntos	0 puntos
C06. Code Organization	Se organiza el código en paquetes aplicando el estándar dns invertido, agrupando las clases según propósito ó funcionalidad (por ejemplo, activities, models, utilities)	No se aplica el estándar dns invertido para los paquetes, pero agrupa las clases según propósito ó funcionalidad.	No se evidencia un criterio de organización del código en paquetes.
	1.5 punto	1.0 punto	0 puntos

C07. Code Quality	Utiliza el lenguaje de programación Dart. La codificación tiene un estilo claro, indentando los bloques de código según los estándares de programación correspondientes al lenguaje, aplicando una lógica consistente en los métodos, condicionales sin escenarios no contemplados, uso adecuado de reutilización de código para evitar redundancia, comentando secciones de código relevantes. Aplica patrones de arquitectura y patrones de diseño.	Utiliza el lenguaje de programación Dart. La codificación es funcional, pero sólo aplica parcialmente los estándares de indentación de bloques de código, ó existen ineficiencias en la codificación: redundancia ó inconsistencias en la lógica de programación, ausencia de comentarios. Aplica parcialmente patrones de arquitectura y patrones de diseño.	No utiliza el lenguaje de programación Kotlin ó la codificación es funcional pero no se evidencia aplicación de estándares ó criterios de eficiencia en la codificación, con ausencia de comentarios, ó no aplica patrones de arquitectura ni patrones de diseño.
	1.5 puntos	1.0 punto	0 puntos
C08. Naming Standards	El desarrollador aplica en todos los nombres de objetos de programación como paquetes, clases, objetos, variables, constantes y métodos la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, así como elementos de UI y recursos.	El desarrollador aplica sólo en parte la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, así como elementos de UI y recursos.	El desarrollador no aplica nomenclatura en inglés para los objetos de programación ó identificadores de elementos de UI y recursos.
	1.0 punto	0.5 puntos	0 puntos
C09. Application Icon	La aplicación tiene asociado un ícono vinculado al propósito de la aplicación.	La aplicación tiene asociado un ícono, pero no se llega a mostrar.	La aplicación sólo tiene asociado el ícono asignado por defecto.
	1.0 punto	0.5 puntos	0 puntos
Total	20 puntos	10.5 puntos	0 puntos

Penalidad

No baja el peso del archivo: -2 pts