

Primer Sprint Trabajo Final

Sirius

Protectora animal

Selene González Curbelo

Mariana Surós Álvarez

26/11/2023



ULPGC

Escuela de
Ingeniería Informática



Índice

Introducción	2
Planificación del Sprint.....	2
Historias de Usuario	2
Enlace al repositorio.....	4
Resumen técnico.....	4
AnimalDao.kt.....	4
NewsDao.kt	5
SiriusDatabase.kt.....	5
Animal.kt	5
News.kt.....	5
User.kt	5
Destinations.kt	5
Color.kt.....	5
Navbar.kt.....	5
AnimalsGallery.kt	6
AnimalInfo.kt.....	6
DonationsScreen.kt	7
HomeScreen.kt.....	7
LandingPage.kt	8
AnimalViewModel.kt.....	9
NewsViewModel.kt	9
UserViewModel.kt.....	9
MainActivity.kt	9

Introducción

Este informe se centra en la planificación y desarrollo del sprint para la aplicación Sirius, albergue animal. El objetivo principal es abordar las funcionalidades clave identificadas a través de historias de usuario.

Planificación del Sprint

Historias de Usuario

ID	1	Nombre	Home Personalizado	
Descripción	Como usuario, quiero una página de inicio personalizada que muestre animales, noticias y otra información importante.			
Criterios de validación	<ul style="list-style-type: none">La página de inicio debe mostrar los animales de la base de datos ordenados por el tiempo que llevan en el refugio.Debe haber un scroll horizontal para cada item de la página de inicio.La página de inicio debe mostrar las noticias de la base de datos ordenadas por la fecha de publicación.	Prioridad	Alta	
		Estimación	3 horas	
		Asignación	Mariana	

ID	2	Nombre	Navbar Personalizado	
Descripción	Como usuario, deseo un navbar para poder navegar por todas las pantallas importantes de la aplicación.			
Criterios de validación	<ul style="list-style-type: none">El navbar debe contener los iconos especificados en el diseño de Figma.Al seleccionar una sección del navbar, los colores y fondos deben actualizarse de acuerdo al diseño.La navegación entre secciones debe ser fluida y reflejar visualmente la selección del usuario.	Prioridad	Alta	
		Estimación	4 horas	
		Asignación	Selene	

ID	3	Nombre	Landing Page	
Descripción	Como usuario, quiero una página de aterrizaje que me de la bienvenida a la aplicación y me proporcione información inicial sobre sus funciones.			
Criterios de validación	<ul style="list-style-type: none">La landing page debe tener un diseño atractivo y acogedor.Debe proporcionar información sobre las funciones claves de la aplicación.Debe incluir llamadas a la acción que inviten al usuario a explorar más.	Prioridad	Baja	
		Estimación	30 minutos	
		Asignación	Selene	

ID	4	Nombre	Galería de animales personalizada	
----	---	--------	-----------------------------------	--

Descripción	Como usuario, quiero una galería de animales personalizada que incluya filtros funcionales y cambios visuales.		
Criterios de validación	<ul style="list-style-type: none"> La galería debe mostrar animales con la información de la base de datos. Se deben agregar filtros funcionales para categorizar y buscar animales. Debe haber cambios visuales al aplicar filtros o al interactuar con la galería. 	Prioridad	Alta
		Estimación	6 horas
		Asignación	Selene

ID	5	Nombre	Base de datos actualizada	
Descripción	Como desarrollador, quiero tener las bases de datos necesarias (news, users) creadas.			
Criterios de validación	<ul style="list-style-type: none">Se creará un script de Python que se ejecute sin errores y genere las bases de datos y realice las inserciones iniciales.Las bases de datos news y users deben crearse.Las bases de datos deberán tener la estructura adecuada para que no falle con Room en Kotlin.	Prioridad	Alta	
		Estimación	4 horas	
		Asignación	Mariana	

ID	6	Nombre	Scroll vertical	
Descripción	Como usuario, quiero poder hacer scroll vertical en las páginas de inicio y galería de animales para explorar más contenido.			
Criterios de validación	<ul style="list-style-type: none">Se debe permitir el scroll vertical en la página de inicio.Se debe permitir el scroll vertical en la página de galería de animales.		Prioridad	Alta
			Estimación	15 minutos
			Asignación	Selene & Mariana

ID	7	Nombre	Página de donaciones	
Descripción	Como usuario interesado en apoyar a la asociación, quiero poder ver una página de donaciones con botones para explorar las opciones de contribución.			
Criterios de validación	<ul style="list-style-type: none">• Debe haber una página de donaciones con un diseño atractivo e intuitivo.• Se deben agregar botones no funcionales – la funcionalidad de pago/donación no se incorporará al ser un refugio ficticio.		Prioridad	Baja
			Estimación	1 hora
			Asignación	Mariana

ID	8	Nombre	Cambios en la base de datos Animal
Descripción	Como desarrollador, quiero realizar cambios en la base de datos Animal para mejoras futuras.		

Criterios de validación	<ul style="list-style-type: none"> • Añadir la columna <i>foster_care</i> como entero que defina si un animal está en casa de acogida o no. • La columna estado debe ser modificado de <i>state</i> a <i>waiting_adoption</i> para definir si un animal está en adopción o en preadopción. • La columna <i>typeAnimal</i> pasa a ser un enumerado de la clase <i>TypeAnimal</i>. • La columna <i>age</i> debe ser modificado a <i>birth_date</i>, para definir la edad a partir de la fecha de nacimiento. • La columna <i>time_shelter</i> debe ser modificado a <i>entry_date</i>, para definir el tiempo en el albergue. 	Prioridad	Media
		Estimación	3 horas
		Asignación	Mariana & Selene

ID	9	Nombre	Información extra de un animal
Descripción	Como usuario interesado en adoptar o acoger, quiero obtener más información sobre un animal al hacer clic en su tarjeta en la galería de animales.		
Criterios de validación	<ul style="list-style-type: none"> • Al hacer clic en la tarjeta de un animal, se debe navegar a una página que muestre información detallada sobre el animal. • La información detallada de un animal debe incluir foto, nombre, edad, raza, descripción, etc. 	Prioridad	Alta
		Estimación	8 horas
		Asignación	Selene & Mariana

Enlace al repositorio

El proyecto *Sirius* se encuentra disponible en el repositorio de GitHub. Dentro del directorio *Main Project* se encuentra el proyecto de Android Studio denominado *Sirius*, así como este informe en formato PDF. El repositorio también incluye un GIF que demuestra el funcionamiento de la aplicación.

<https://github.com/mele13/PAMN/tree/main/Main%20Project>

Resumen técnico

Presentación del resumen técnico de los archivos relevantes del código fuente.

AnimalDao.kt

Define la interfaz *AnimalDao* para realizar operaciones en la base de datos de animales, como inserción, actualización y consulta. Se ha añadido la función *getAnimalById* para obtener un animal por su ID, la cual se utiliza en la sección *AnimalInfo*.

NewsDao.kt

Define la interfaz *NewsDao* para realizar operaciones en la base de datos de noticias, como inserción, actualización y consulta.

SiriusDatabase.kt

Define la clase *SiriusDatabase*, que representa la base de datos de la aplicación. Implementa la creación de la base de datos utilizando la biblioteca Room. Se ha cambiado el nombre del archivo de *AnimalDatabase* a *SiriusDatabase*, se ha añadido el path de la base de datos y se ha incorporado la nueva base de datos mediante *createFromAsset*. Además, implementa el patrón Singleton para garantizar una única instancia de la base de datos.

Animal.kt

Define la entidad de datos *Animal* con sus atributos: nombre, edad, descripción, entre otros. Se ha realizado una modificación en los atributos:

- *state* -> pasa a ser *waiting_adoption*, ahora de tipo entero, indicando si un animal está en adopción (1) o preadopción (0).
- *type_animal* -> pasa a referenciar al enumerado *TypeAnimal*.
- *age* -> pasa a ser *birth_date*, ahora de tipo text, indicando la fecha de nacimiento con formato yyyy-mm-dd.
- *time_shelter* -> pasa a ser *entry_date*, ahora de tipo text, indicando la fecha de nacimiento con formato yyyy-mm-dd.

Además, se ha añadido el atributo *foster_care*, de tipo entero, que indica si un animal se encuentra en casa de acogida (1) o no (0). También, el atributo *photo_animal*, de tipo text, que indica la ruta de la imagen dentro del proyecto.

News.kt

Define la entidad de datos *News* con sus atributos: título, descripción, fecha de publicación, entre otros.

User.kt

Define la entidad de datos *User* con sus atributos: nombre de usuario, email, contraseña, rol, entre otros.

Destinations.kt

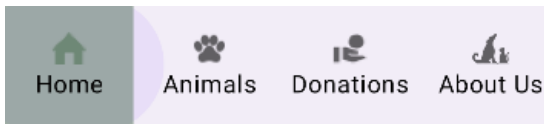
Define las rutas y destinos de la aplicación. Se han creado funciones para generar destinos y una lista de destinos. La estructura de datos guarda información sobre la ruta, los iconos seleccionados y no seleccionados, así como el identificador del texto del icono. Se ha añadido la ruta *AnimalInfo* para la información detallada de los animales.

Color.kt

Define constantes de colores personalizados utilizados en la interfaz de usuario de la aplicación.

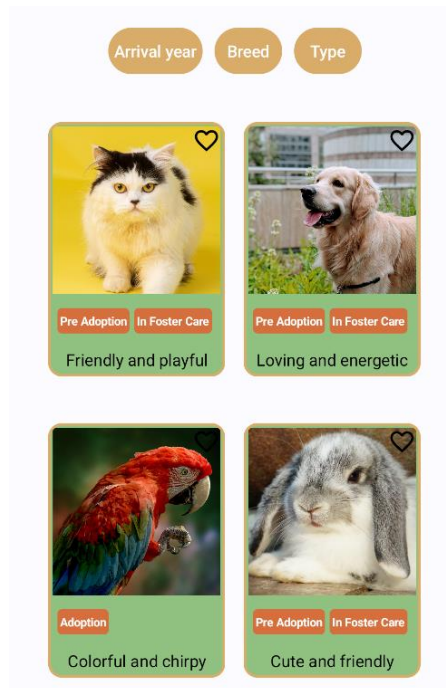
Navbar.kt

Define la barra de navegación de la aplicación, que muestra los iconos y textos correspondientes a diferentes secciones de la aplicación.



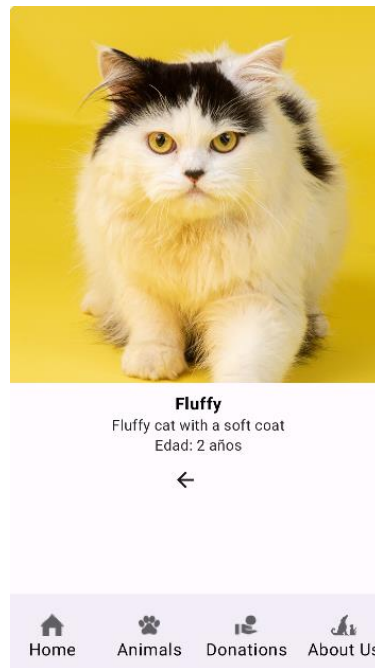
[AnimalsGallery.kt](#)

Define la pantalla de la galería de animales. La galería utiliza una disposición en forma de lista que presenta tarjetas individuales para cada animal disponible para adopción. Cada tarjeta incluye una imagen y el nombre y edad del animal. Se ha añadido filtros por edad, raza y tipo. Se ha cambiado la parte visual para hacer referencia al diseño de Figma.



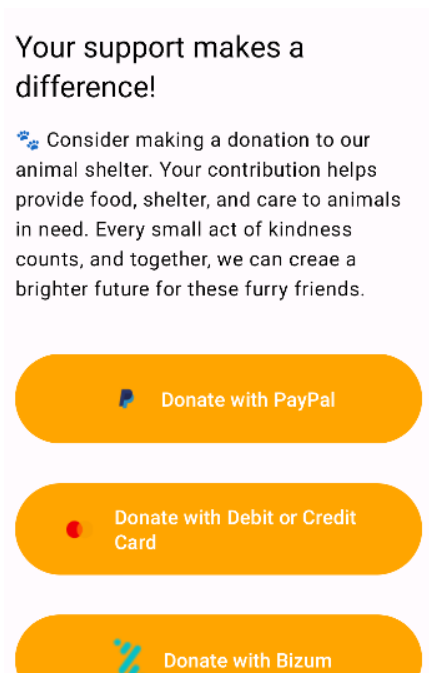
[AnimalInfo.kt](#)

Define la pantalla sencilla de información detallada de un animal, mostrando su imagen, su información y un botón de retroceso.



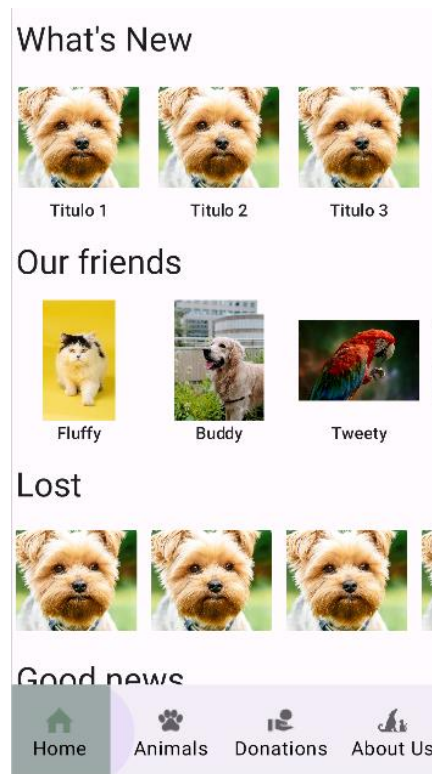
DonationsScreen.kt

Define la pantalla de donaciones con botones (no funcionales) para donar a través de PayPal, tarjeta de crédito/débito y Bizum.



HomeScreen.kt

Representa la pantalla principal de la aplicación, mostrando las noticias y los animales de la base de datos.



LandingPage.kt

Define la pantalla de inicio de la aplicación con una imagen de fondo, el logotipo de Sirius y botones para iniciar sesión o unirse como invitado.



[AnimalViewModel.kt](#)

Implementa la clase *AnimalViewModel*, que actúa como intermediario entre la base de datos de animales y la interfaz de usuario. Proporciona métodos para obtener datos de animales. Se han añadido nuevas funciones para obtener un animal por su ID, funciones de filtro, entre otras.

[NewsViewModel.kt](#)

Implementa la clase *NewsViewModel*, que actúa como intermediario entre la base de datos de noticias y la interfaz de usuario. Proporciona métodos para obtener datos de noticias.

[UserViewModel.kt](#)

Implementa la clase *UserViewModel*, que actúa como intermediario entre la base de datos de usuarios y la interfaz de usuario. Proporciona métodos para obtener datos de usuarios.

[MainActivity.kt](#)

Define la actividad principal de la aplicación. Se ha eliminado la inicialización del *ViewModel*.