

ARMADEIRA

Network data collector

Mobile App User's Guide
Armadeira

Contents

Contents.....	2
1. Introduction.....	3
1.1 Purpose and scope.....	3
1.2 How/What kind of information is collected?.....	3
1.3 Compatibility.....	3
1.4 Organization.....	3
2. Technical Specification.....	4
3. Installation.....	5
4. App overview.....	6
4.1 Splash screen.....	6
4.2 Permissions screen.....	7
4.3 Home screen.....	11
4.4 Device screen.....	12
4.5 Memory screen.....	13
4.6 Location screen.....	14
5. How to use.....	15
6. Description of application's functions.....	18
7. Additional information about the collected data.....	19
7.1 Device.....	19
7.2 Location.....	19
7.3 Wifi.....	19
7.4 Mobile.....	21
7.5 Memory.....	23
7.6 Battery.....	23
7.8 Bluetooth.....	24
8. Release Notes.....	25
8.1 Screen Layout.....	25
8.2 Data Storage.....	26
8.3 App Notification.....	26
8.4 Code Structure Changes.....	26

1. Introduction

1.1 Purpose and scope

The Armadeira app was developed for mobile devices (smartphones, tablets, or others) with the purpose of collecting information about the network to which the device is connected. It works by collecting network communication data. The name "Armadeira" was chosen for the app because it primarily collects wireless network data, and its network structure is similar to a spider web. Furthermore, the team wanted a name that was related to the Amazon region. The word "Armadeira" refers to a species of spider found in the Amazon region, and therefore, was considered an appropriate choice.

1.2 How/What kind of information is collected?

The Armadeira app's main objective is to collect network data on Android smartphones. This data includes Wi-Fi and mobile network information and is collected at a granularity of one second. To ensure user security and privacy, Armadeira stores the collected data in a .csv file on the device itself. This ensures that the data is not shared with third parties and that users have full control over their information. With the collected data, machine learning techniques can be used to analyze performance and predict uplink and downlink transfer rates, identify possible bottlenecks, and help users make informed decisions about their Internet connection.

1.3 Compatibility

To use the Armadeira application, you must have a smartphone with an Android operating system, at least version 10 of the operating system. In addition, your device must have at least 2 GB of RAM and 150 MB of free storage space (recommended). These requirements are essential to ensure the proper functioning of the application and to avoid performance and stability issues. Therefore, it is important to check device compatibility before downloading and installing the app in order to have a satisfactory experience.

1.4 Organization

This application was developed by the Intelligent Hardware (IH) group, part of the project characterized as Research, Development and Technological Innovation (RD&I), entitled Artificial Intelligence Techniques for Software Performance Analysis and Optimization (SWPERFI). Advanced data mining and artificial intelligence techniques, including deep learning and general computing, are being applied to analyze performance metrics. Specifically, the project is developing innovative methods to verify dependencies, establish correlations, determine possible problems, and create a new approach embedded in a prototype tool for verification, testing, and optimizing software performance using AI techniques. Any questions, concerns, or issues regarding the app may be resolved by contacting the SWPERFI project, and they will direct you to the IH team.

SWPERFI website: <https://swperfi.icomp.ufam.edu.br/>

General Info Contact: swperfi@icomp.ufam.edu.br

Armadeira App: E-mail: swperfi-ih@icomp.ufam.edu.br

2. Technical Specification

Name	Armadeira
Description	The Armadeira app's main objective is to collect network data on Android smartphones.
Performed by	SWPERFI
Responsible team	Intelligent Hardware - IH
Support	You must have a smartphone with an Android operating system, at least version 10 of the operating system. In addition, your device must have at least 2 GB of RAM and 150 MB of free storage space (recommended).
Current version	3.00.00-alpha
Download link	Armadeira App (http://swperfi-project.github.io/Pages-dev/ArmdNetworkDataCollector-app)

3. Installation

Note: In this new version, if you have the previous version of the Armadeira application installed, you will need to uninstall the application and remove the "/Armadeira" folder located at: InternalStorage/Documents/Armadeira

To install the Armadeira app on an Android device, follow the steps below:

1. Access the app file, which is stored on GitHub Pages, at the following link: Armadeira App (<http://swperfi-project.github.io/Pages-dev/ArmdNetworkDataCollector-app>);
2. The file is already available for public download, so it is not necessary to request permission to download it;
3. Download the application file on the Android device directly from the Google Drive app or using a web browser, once you have access to the file;
4. Make sure to allow the download of unknown files on your Android device by enabling the "Unknown sources" option in the security settings.
5. After the application file is downloaded, access the device's file management application and locate the Armadeira application file as illustrated in Figure 1;
6. Touch the file to start the installation process;
7. The device may display a security warning message informing that you are about to install an app from an unknown source. As the app comes from a secure source, tap on "Install" to continue;
8. The installation of the application may take a few seconds, and after completion, you can open it from the device's app menu.

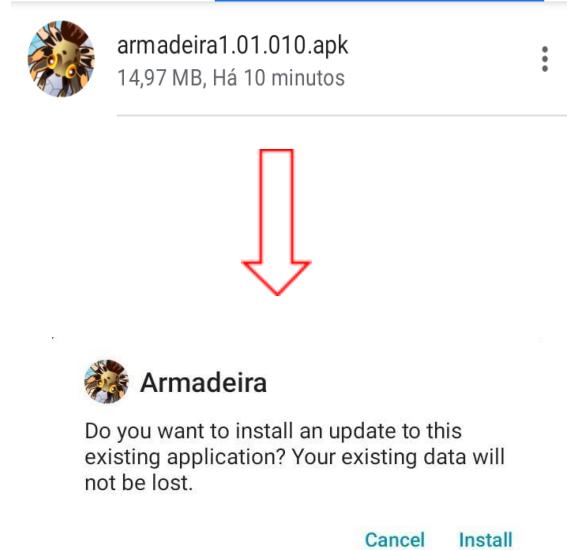


Figura 1. Example of the installer obtained through the link in this manual.

4. App overview

4.1 Splash screen

The application displays a splash screen that shows the application logo and loads some project settings.

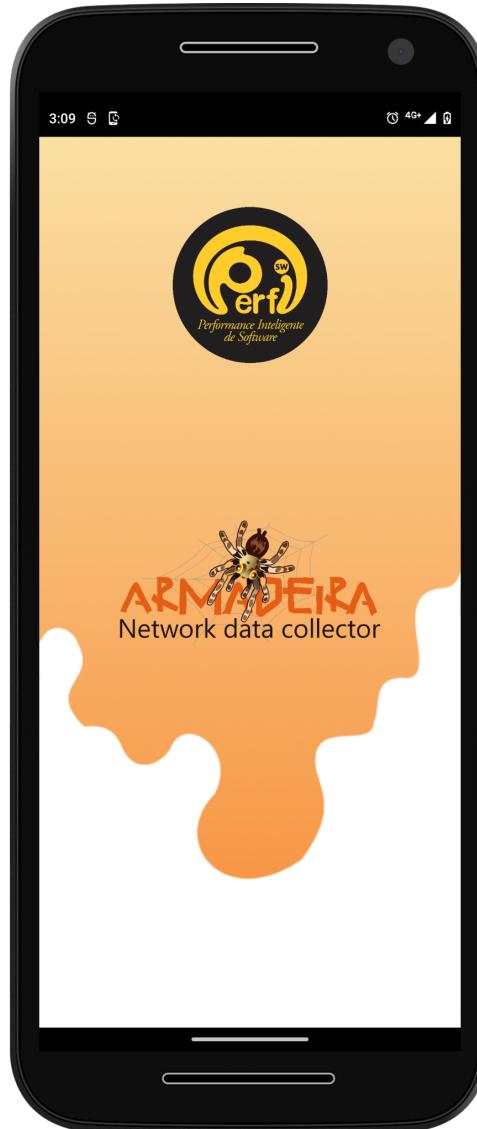


Figura 2. Loading screen (splash screen) of the Armadeira application.

4.2 Permissions screen

After the splash screen, the app is directed to a welcome screen, followed by a screen briefly explaining the purpose of the app and informing that it will be necessary to request certain permissions in order for the app to fulfill its purpose, as illustrated in Figure 3.

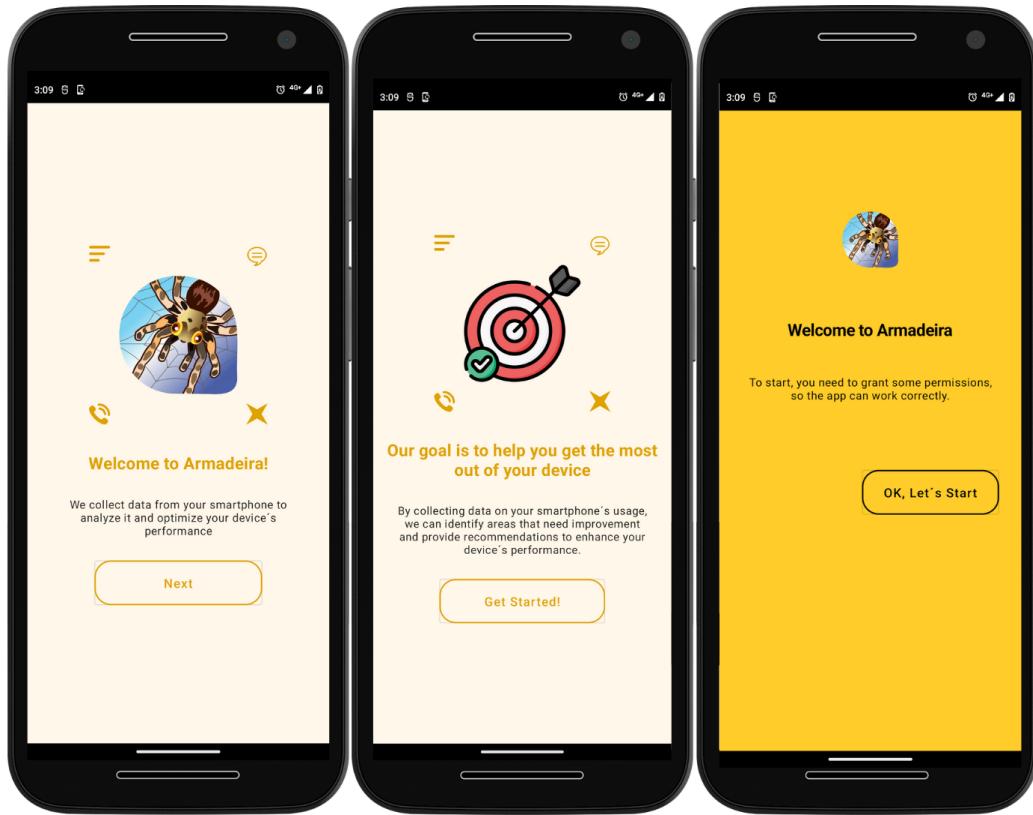


Figura 3. Initial screens welcoming the user, stating the app's purpose, and informing about permission requests.

In Figure 4, we have the first permission on the location screen 1, this permission can be granted by clicking on "During app use". The next permission, shown in Figure 4, is intended to assist in reading and writing CSV files generated by Armadeira on newer versions of Android. Simply activate the "Access to manage all files" option and go back to the previous screen.

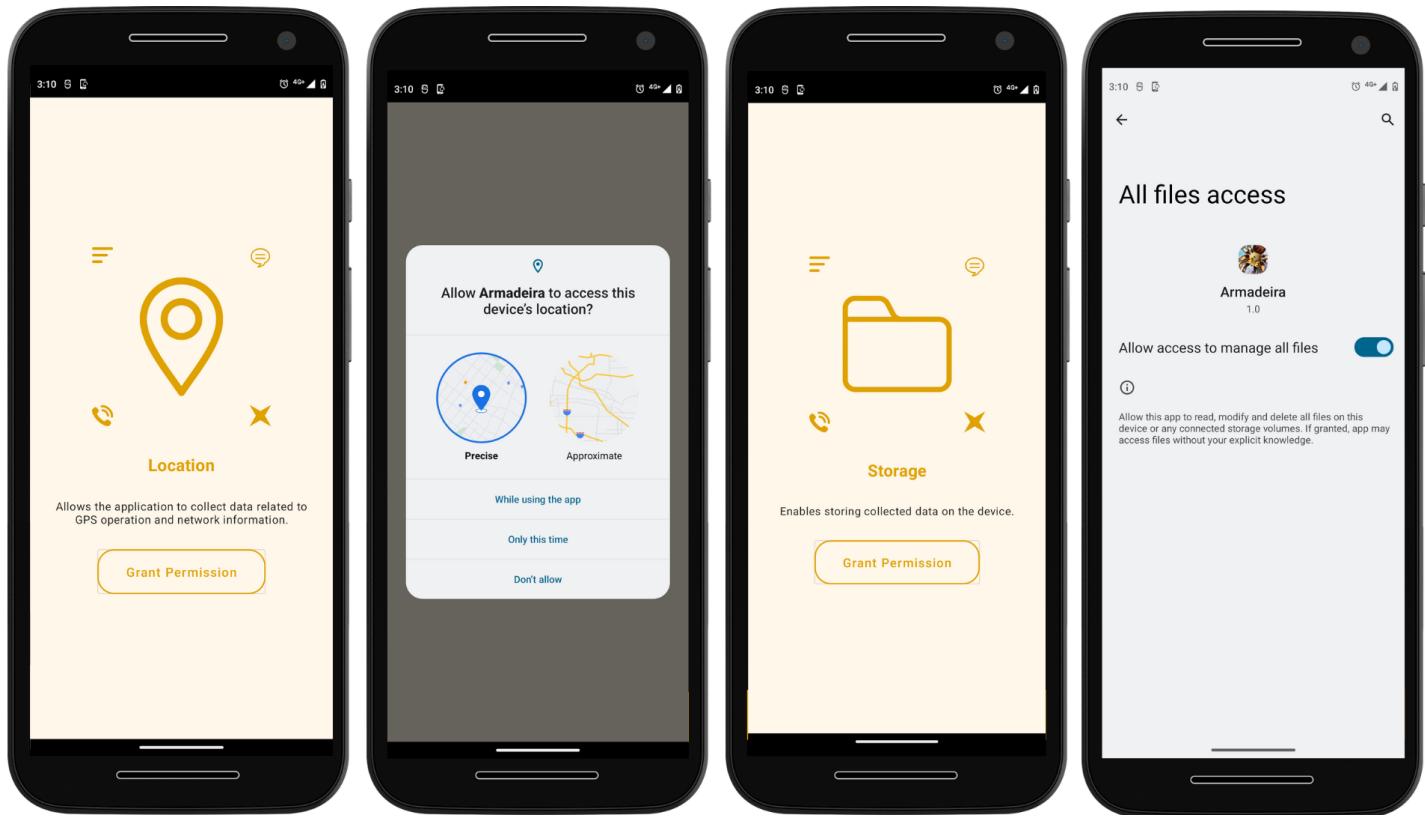
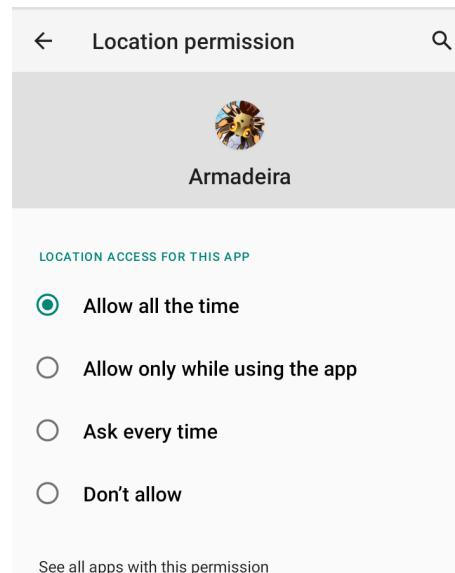


Figura 4. Required permissions for: a) Obtaining location; b) Accessing device files.

OBS: to correct collect of location, it is necessary to give the permission to “Allow all the time” in the “Settings”



In Figure 5, the first permission requested allows collecting information about mobile networks. The other permission, shown in Figure 5, requires a bit more time. The steps are to click "OK" on the message that appears on the app screen. After that, a new screen titled "Overlay other apps" will appear, with a list of apps. In this list, you need to locate the "Armadeira" app, possibly having to scroll to the end of the list. Once you find the app name, simply click on it and accept the overlay permission. Once the process is complete, just go back until you reach the Armadeira screen again.

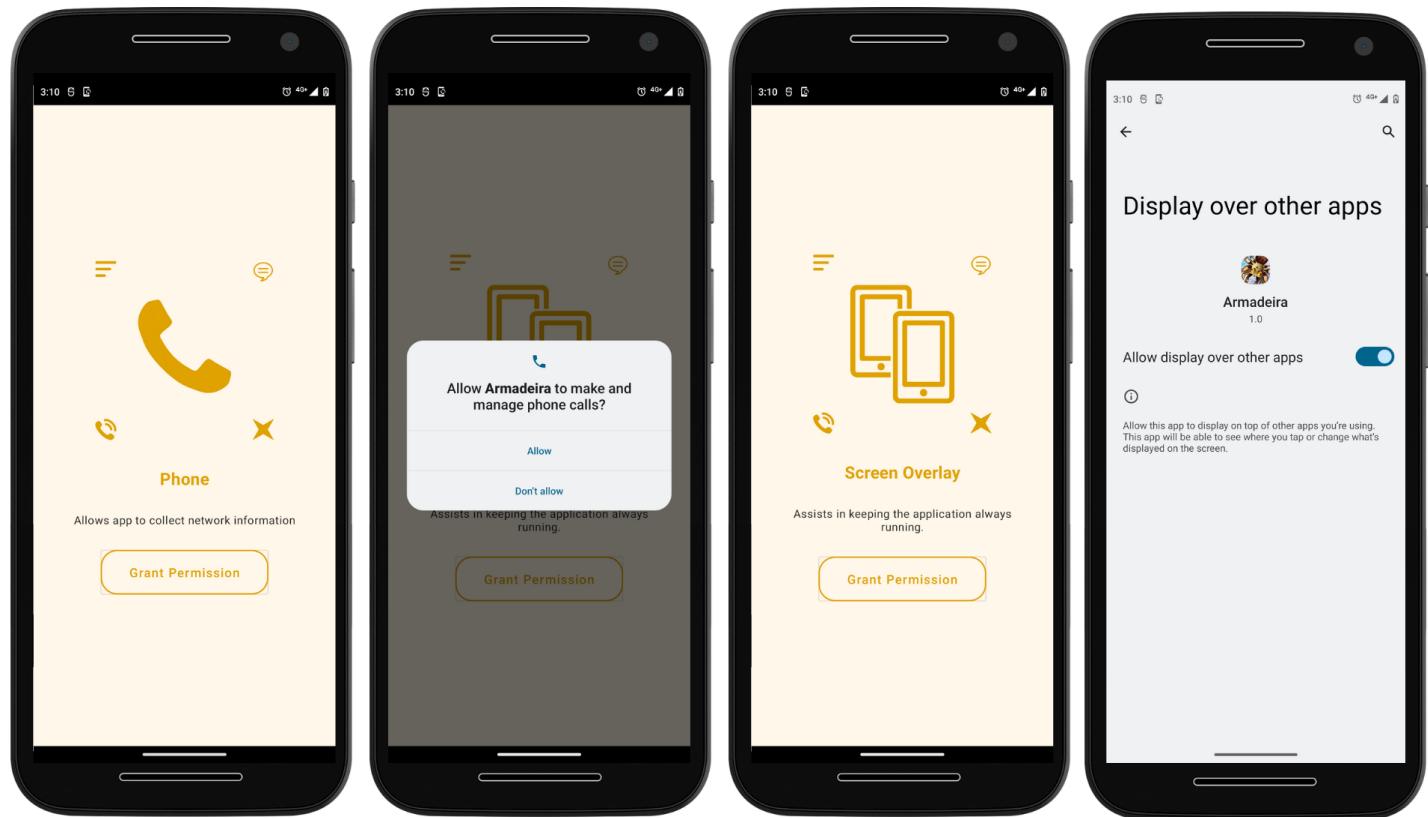


Figura 5. Required permissions for: a) Accessing information about mobile networks; and b) Running Tucandeira app in the background.

Finally, a new tab of "Required permission" will appear, where you just need to click "Ok" and "Allow access to usage" before returning to the app. Once the process is completed, the user can click on the "Start App" button at the end of the screen to access the main screen of the application, as shown in Figure 6.

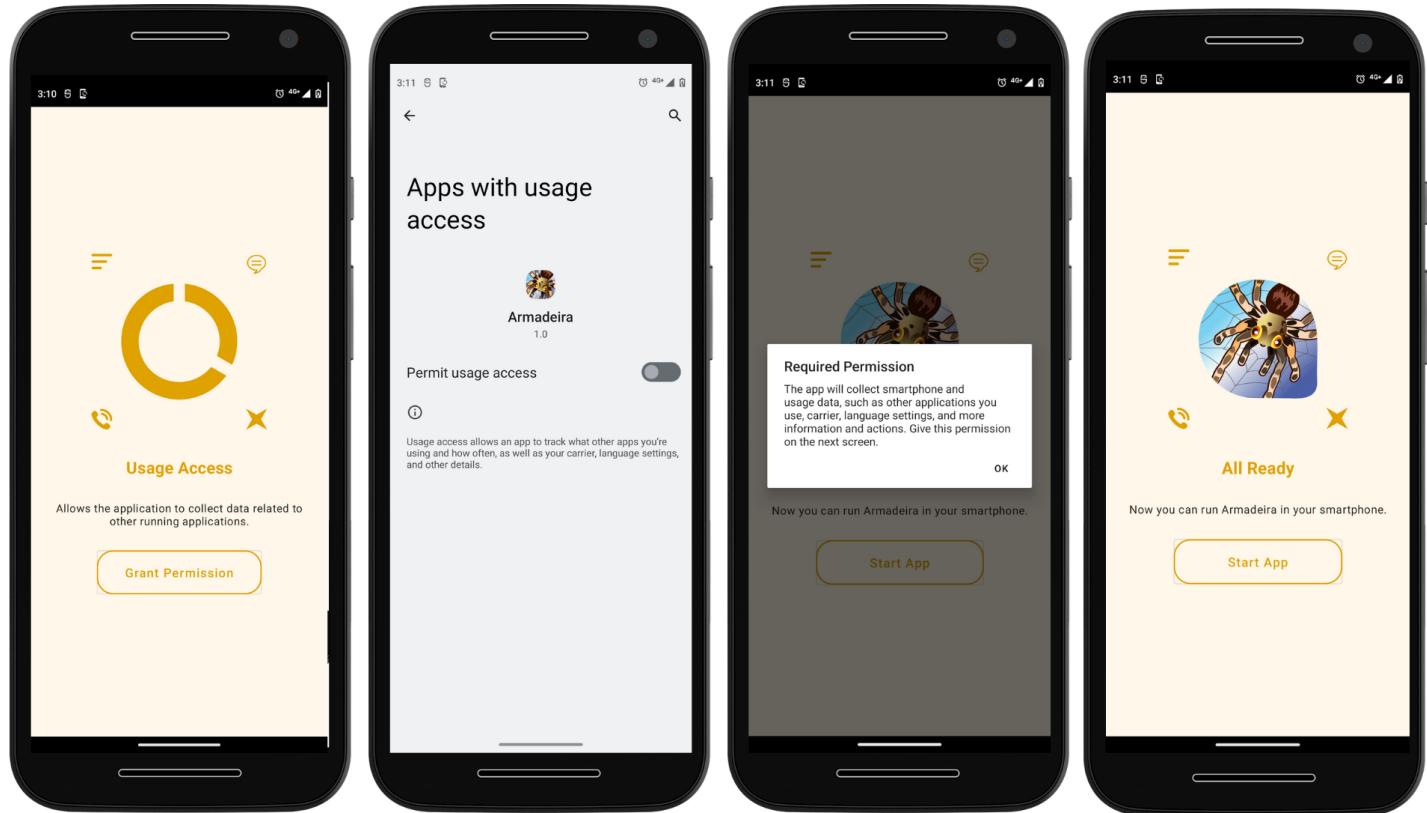


Figura 6. On the first screen, permission to manage files for reading and writing data in ROM memory. On the fourth screen, it is possible to start the app.

4.3 Home screen

The Home screen displays some information such as the Android version that the device is using, as well as the device model. As for the Network information, the application informs whether the device has Wi-Fi, mobile data, and Bluetooth enabled or not. Additionally, it checks whether the device has location services enabled.

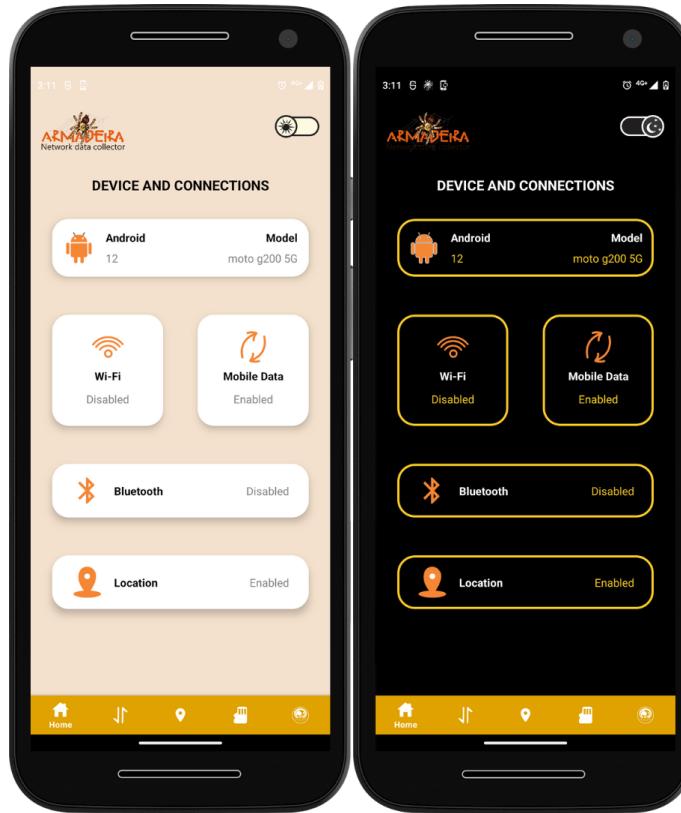


Figura 7. Home screen.

4.4 Device screen

On the Device screen, the application shows which signal provider the device is using, as well as the type of connection used. It also displays the network band used by the device and shows the throughput that occurs on the network. It also displays the frequency and signal strength.

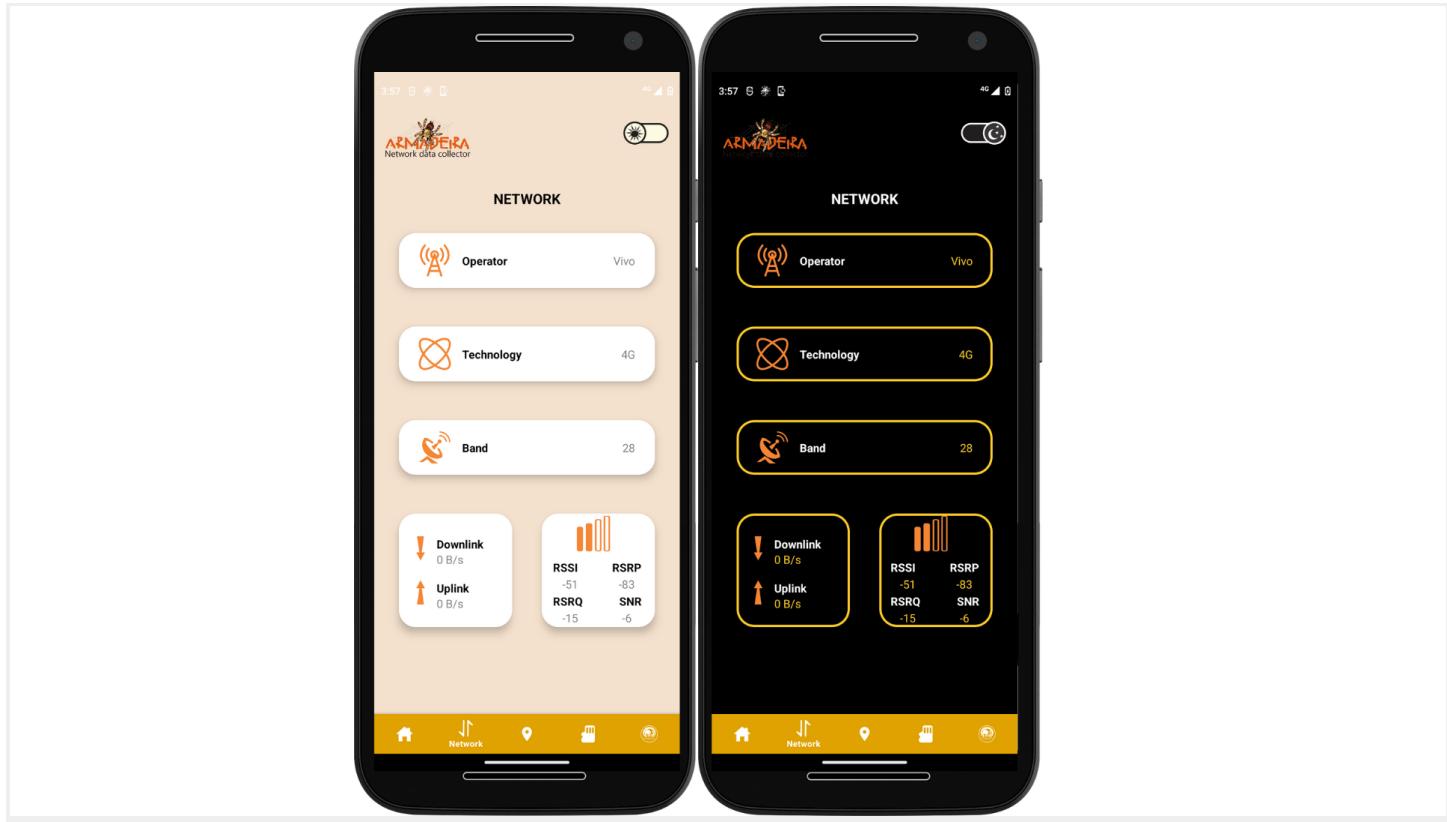


Figura 8. Device screen.

4.5 Memory screen

The Memory screen shows the total space available in the device's internal storage as well as the amount of memory that is already being used. It also displays the amount of RAM memory in the device and the amount of memory that is being used.

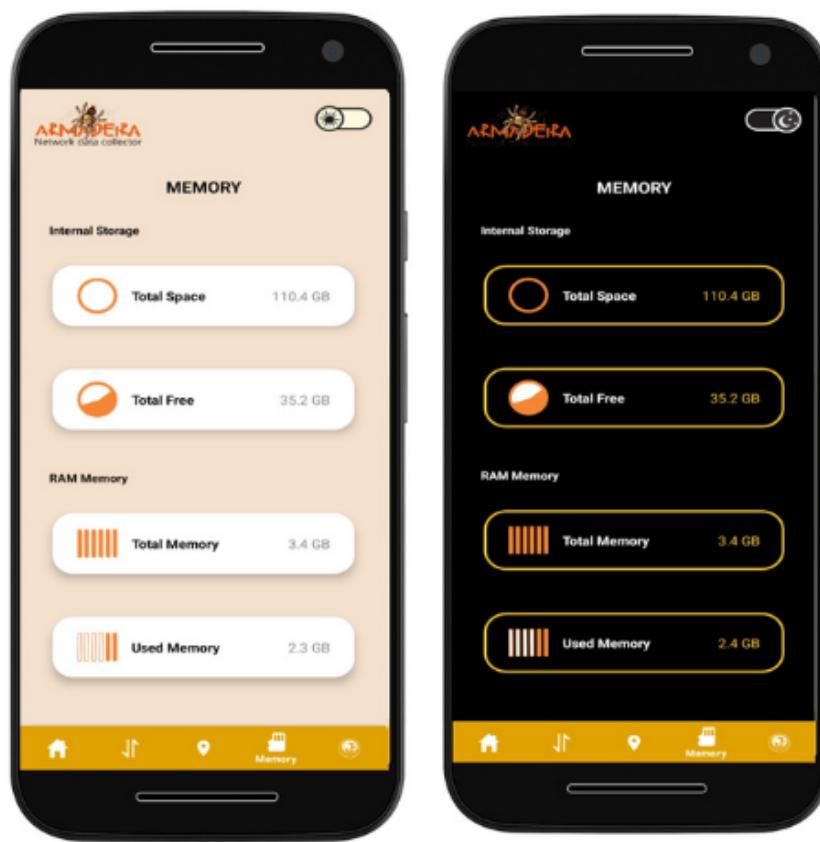


Figura 9. Memory screen.

4.6 Location screen

In this latest update, a new feature has been introduced: a dedicated screen for displaying the user's location. Now, when accessing this new version, users will be able to view the map clearly and accurately, with their current location highlighted. This addition will provide a more intuitive and practical experience when using our application.

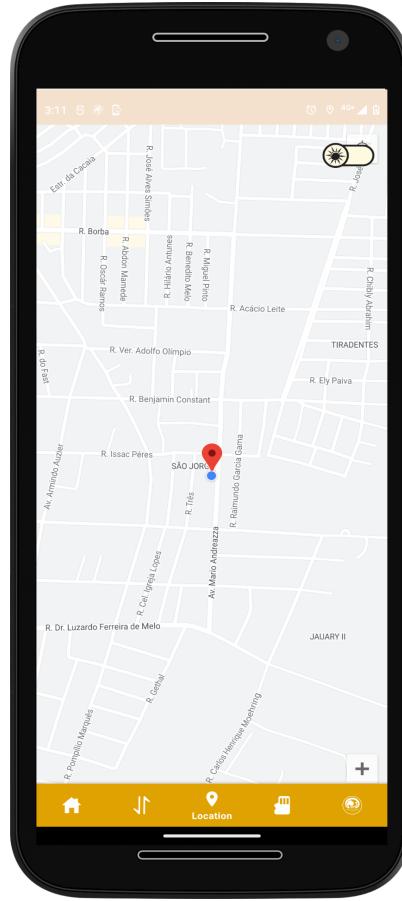
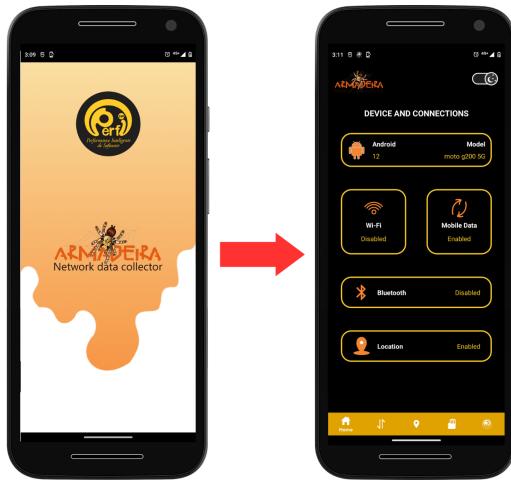


Figura 10. Location screen.

5. How to use

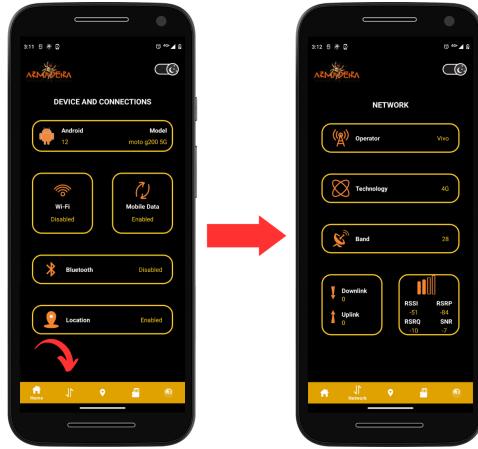
To use the Armadeira Network Data Collector application, you need to select the application by clicking on the app icon in your device. The app will open in a splash screen and in sequence will show you the home screen, as shown in Figure 11.



- ❖ In the home screen you can see the information about:
 - See if the Wi-fi, Mobile Data, Bluetooth and Location are enabled or not .
 - Operational system version
 - Model's device
 - Network information

Figura 11. How to access the home screen.

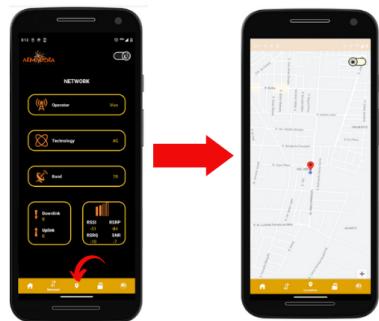
To see more information about the Network, you need to click on the Device icon, to move to the Device Screen, as shown in Figure 12.



- ❖ In the device screen you can see information about:
 - Operator
 - Technology
 - Band
 - Throughput
 - Frequency
 - Intensity

Figura 12. How to access the device screen.

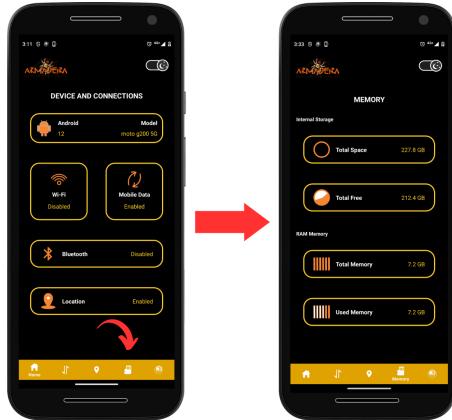
To see more information about the current location of the device, you need to click on the location icon and you will move to the Location Screen, as shown in Figure 13.



- ❖ In the location screen you can see the current location of your mobile device

Figura 13. How to access the location screen.

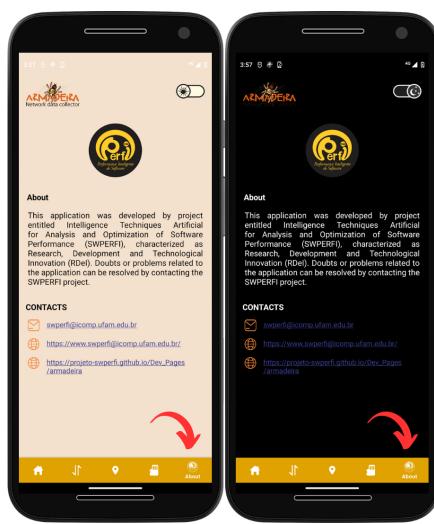
To see more information about the memory of the device, you need to click on the memory icon and you will move to the Memory Screen, as shown in Figure 14.



- ❖ In the Memory Screen you can see information about:
 - the internal storage
 - RAM memory

Figura 14. How to access the memory screen.

To see more information about the app and the developer group, you need to click on the SWPerfi icon and you will move to the About Screen, as shown in Figure 15.



- ❖ In the About Screen you can see information about the Project that developed the app, and some ways to contact and know more about the project.

Figura 15. How to access the About screen.

6. Description of application's functions

The Armadeira offers several functionalities that aim to collect and store data from mobile devices, mainly related to device connectivity. This data is stored locally on the device itself, in the "Armadeira" folder, and can be accessed through the file manager. One of the main features of Armadeira is real-time data collection, which is updated every second in CSV spreadsheets. This data includes information about the device's internet connection, such as frequency, signal strength, bandwidth, among others. Other important data is also collected throughout the day and stored in the CSV spreadsheet, such as latitude and longitude, amount of RAM being used, amount of free memory on the device, among other information.

7. Additional information about the collected data

7.1 Device

Feature name	Type	Value	Measure	Description
deviceId	string	string	no	A unique identifier assigned to a specific device.
date	string	Ex. 01/04/2023	no	Indicates the date the data was generated or recorded
time	string	Ex. 12:01:21	no	Indicates the date the data was generated or recorded

7.2 Location

Feature name	Type	Value	Measure	Description
locationStatus	boolean	1 or 0	no	(1) indicates if the user's location is enabled, (0) indicates disabled.
latitude	double	floating	no	The latitude value of the user's location.
longitude	double	floating	no	The longitude value of the user's location.

7.3 Wifi

Feature name	Type	Value	Measure	Description
wifiStatus	boolean	1 or 0	no	Indicates the current state of the device's Wi-Fi connection, whether it is connected or disconnected.

downstreamBandwidth	int	floating	Kbps	The download speed of data on a Wi-Fi network, measured in Kbps. This variable can be used to check the available download bandwidth in a Wi-Fi connection.
upstream Bandwidth	int	floating	Kbps	The upload speed of data on a Wi-Fi network, measured in Kbps. This variable can be used to check the available upload bandwidth in a Wi-Fi connection. Both variables can be used to monitor the performance of the Wi-Fi connection and ensure that it meets the user's needs.
ssid	string	string	no	The name of the Wi-Fi network to which the device is connected. It is useful for identifying the network to which the device is associated.
ip	string	Ex. 192.168.1.1	no	The IP address assigned to the device by the Wi-Fi router to which it is connected. It is useful for identifying the device on the network.
signal Level	int	floating	no	A measure of the Wi-Fi signal strength received by the device. The higher the value, the better the signal quality.
networkIng	int	floating	no	A unique identifier assigned to the Wi-Fi network to which the device is connected.
speed	int	floating	Mbps	The current Wi-Fi connection speed of the device, measured in Mbps (megabits per second).
tx Link Speed	int	floating	Mbps	The current transmission speed of the device's Wi-Fi connection, measured in Mbps.
rxLinkSpeed	int	floating	Mbps	The current reception speed of the device's Wi-Fi connection, measured in Mbps.
wifi Frequency	int	floating	Mhz	The frequency of the Wi-Fi network to which the device is connected, measured in MHz (megahertz).
wifiRssi	int	floating	dbm	A measure of the strength of the Wi-Fi signal received by the device, in dBm (decibels relative to the power of one milliwatt). The higher the value, the better the signal quality.
wifiWap	string	Ex. WEP,	no	The Wi-Fi security protocol used by the

		WPA, WPA2		network to which the device is connected, such as WEP, WPA, or WPA2.
maxSupportedTxLinkSpeed	int	floating	Mbps	The maximum transmission speed supported by the device's Wi-Fi connection, measured in Mbps.
maxSupportedRxLinkSpeed	int	floating	Mbps	The maximum reception speed supported by the device's Wi-Fi connection, measured in Mbps.
wifiStandard	int	floating	no	The Wi-Fi standard supported by the network to which the device is connected, such as 802.11ac or 802.11n.

7.4 Mobile

Feature name	Type	Value	Measure	Description
mobileStatus	boolean	1 or 0	no	(1) indicates if the mobile device is turned on, (0) indicates turned off.
netType	string	1G, 2G, 3G, 4G or 5G	no	A parameter that indicates the type of mobile network currently in use by the mobile device.
roaming	bool		no	Indicates whether the device is roaming on a network outside of its home network.
downstreamBandwidth	int	floating	Kbps	The download speed of data on a mobile network, measured in kilobits per second (Kbps).
upstreamBandwidth	int	floating	Kbps	The upload speed of data on a mobile network, measured in Kbps.
rx	long	floating	Bytes	Received data rate in Bytes.
tx	long	floating	Bytes	Transmitted data rate in Bytes.
networkOperator	string	Ex. “Vivo”, “Claro”, “Tim”	no	Identifies the mobile network operator to which the mobile device is connected.
chipOperator	string	Ex. “Vivo”, “Claro”, “Tim”	no	Identifies the operator of the SIM card in the mobile device.

mcc	string	Ex. 724	no	Mobile country code (MCC) - a unique identifier assigned to a country for mobile network purposes.
mnc	string	Ex. 11	no	Mobile network code (MNC) - a unique identifier assigned to a mobile network within a country.
asuLevel	int	Ex. 32	no	Absolute signal strength value in GSM (ASU) units.
cqi	int	floating	no	Channel quality indicator - a measure of the quality of the communication channel.
dbm	int	floating	dbm	Received signal power in decibels relative to one milliwatt (dBm).
level	int	floating	dbm	Received signal strength indicator (RSSI) in dBm.
rsrp [dBm]	int	floating	dbm	Reference signal received power (RSRP) in dBm.
rsrq	int	floating	db	Reference signal received quality (RSRQ) in dB.
rssi [dBm]	int	floating	dbm	Received signal strength indicator (RSSI) in dBm.
rssnr	int	floating	db	Reference signal received quality (RSRQ) in dB.
earfcn	int	floating	no	E-UTRA Absolute Radio Frequency Channel Number (EARFCN) - a unique identifier for a radio channel in a cellular network.
band	int	floating	no	The frequency band on which the mobile device is operating.
frequency [MHz]	int	floating	Mhz	The frequency in MHz at which the mobile device is operating on a cellular network.
ci	int	floating	no	Cell identity (CI) of the current mobile network cell to which the device is connected.
pci	int	floating	no	Physical cell identity (PCI) of the current mobile network cell to which the device is connected.
tag	int	floating	no	A user-defined label or tag assigned to the

				data.
downLinkFrequency	int	floating	Mhz	uplink frequency channel
upLinkFrequency	int	floating	Mhz	downlink frequency channel

7.5 Memory

Feature name	Type	Value	Measure	Description
availableRAM	long	float	GB	The amount of available random-access memory (RAM) in the device, measured in gigabytes (GB).
criticismRAM	boolean	True or False	no	Indicates if the device's RAM usage has reached a critical level. (True) indicates that the RAM usage is critical, while (False) indicates that it is not.
totalRAM	long	floating	GB	The total amount of RAM installed in the device, measured in gigabytes(GB).
usedRAM	long	floating	GB	The amount of RAM currently being used by the device, measured in gigabytes(GB).
memory	int	floating	GB	The memory capacity of the device, measured in gigabytes (GB).
freeMemory	long	floating	GB	The amount of free memory available in the device, measured in gigabytes (GB).

7.6 Battery

Feature name	Type	Value	Measure	Description
levelBattery	int	0-100	%	Indicates the current battery level of the device, ranging from 0% to 100%.
batteryHealth	int	1-7	no	Indicates the health status of the device's battery, ranging from 1 (very poor health) to 7 (excellent health).
batteryConnectionStatus	int	0-1	no	Indicates whether the device's battery is currently connected or disconnected

batteryChargingStatus	int	0-3	no	Indicates the current charging status of the device's battery, ranging from 0 (not charging) to 3 (fully charged).
batteryTemperature	float	-50 to 150	Celsius	Indicates the current temperature of the device's battery in Celsius.
batteryCurrent	double	floating	mA	Indicates the current flow of electrical charge through the device's battery in mA
batteryVoltage	float	floating	Volts	Indicates the current voltage of the device's battery in volts.
batteryPower	double	floating	Watts	Indicates the current power output of the device's battery in watts.
batteryCapacity	double	floating	mAh	Indicates the current capacity of the device's battery in milliampercere-hours (mAh).
batteryPresence	boolean	1 or 0	no	Indicates whether a battery is present in the device

7.8 Bluetooth

Feature name	Type	Value	Measure	Description
bluetoothStatus	boolean	1 or 0	no	(1) indicates if the Bluetooth is enabled, (0) indicates disabled.

8. Release Notes

This latest update consisted of refactoring all network variables with the aim of improving code readability and maintainability. This included organizing the variables into logical sections and using more descriptive names for each one. This refactoring should make the code easier for other developers to understand and reduce the time it takes to make future changes. In this new version, a new design layout was applied, in order to make the screens smoother for users, with less heavy colors. Below is the section containing the specifications of the changes made in this latest update, highlighting the new design, inclusion of new network information on the app screen, addition of the location screen, changes in data storage for collections, and others that may be checked below:

8.1 Screen Layout

In the latest version of the Armadeira application, we have made a series of significant changes to the user interface. We have **optimized the app's colors for a smoother palette, aiming for greater energy efficiency**. The excessive color saturation from previous versions has been removed, and we have introduced a new dark mode.

Dark theme has been incorporated as the default in the application, offering a more visually appealing and comfortable option for use in low-light environments.

Functionality for toggling between app themes (light or dark mode): In this version, users have the ability to dynamically switch between the available themes in the application, allowing customization according to their own aesthetic preferences.

Permission

In this updated version, we have introduced new interfaces to clarify the purpose of the application and the justification for requesting certain permissions that enable effective data collection by the software. During the interaction with these interfaces, the application requests the user's consent for these permissions.

Home Screen - Device and Network

The layout of the application's home screen has been modified, but the functionalities remain unchanged, including the display of the smartphone model, Android version, status: WiFi, mobile data, Bluetooth, and location.

Mobile Data Screen

The network screen has undergone a significant layout redesign and gained important additions. It now presents uplink and downlink information and displays network strength parameters such as RSSI, RSRP, RSRQ, and SNR. These parameters are crucial for evaluating network quality.

Location Screen

In this latest update, a new feature has been introduced: a dedicated screen for displaying the user's location. Now, upon accessing this new version, users will be able to view the map clearly and accurately, with their current location highlighted. This addition will allow for a more intuitive and practical experience when using our application.

Memory Screen

The changes to the memory screen were purely aesthetic, with updates limited to the layout only.

About Screen

The 'About' screen has been added with the purpose of providing information about the application, developed by the SwPerfi Project, including contact details for any questions or suggestions.

8.2 Data Storage

The data collection storage is done locally in the .csv file format. The CSV file is now available at:

InternalStorage/Documents/Armadeira /Log

Changes have been made to the format in which the data was stored. Now, the data is stored in a single table, meaning a single record, with the aim of optimizing the structure and facilitating faster data analysis.

8.3 App Notification

Now the application features a notification icon to provide users with relevant information and alerts. For example, it can notify users that data is being collected in the background. An icon has been added to the notification as well.

8.4 Code Structure Changes

Introduction of the 'domain' module: In this update, all data collection functions have been encapsulated within the 'domain' module, promoting more structured code and facilitating its maintenance.

Implementation of Dependency Injection via Jetpack: The Jetpack library was used to implement Dependency Injection. This technique allows for greater module decoupling and enhances code testability, making it more robust and less prone to failures.

Refactoring following the MVVM architecture: The code has been restructured to follow the Model-View-ViewModel (MVVM) pattern, enabling a more efficient separation between business logic and user interface, which increases maintainability and scalability of the software.



Adoption of Coroutines in data collection: Coroutines have been employed in the implementation of the data collection process, resulting in a more efficient and responsive application. This feature allows for optimized thread management, avoiding unnecessary blocking.

Introduction of a new service for variable rendering in the interface: A dedicated service has been created, using Coroutines, to render variables in the user interface. This provides a smoother and up-to-date visualization of variables in the application.

Note: In this new version, if you have the previous version of the Armadeira application installed, it will be necessary to uninstall the application and remove the folder "/Armadeira" located at: InternalStorage/Documents/Armadeira.