TocáMexer: Mobile Application to Track Physical Activity in a Clinical Context using GNSS

Bernardo Carapito, Thesis Supervisor - Prof. José Sanguino
bernardo.carapito@tecnico.ulisboa.pt, jose.sanguino@tecnico.ulisboa.pt
Instituto Superior Técnico, Portugal

***Abstract*—Nowadays, physical exercise is considered an effective way to prevent numerous health problems, like cardiovascular diseases, strokes, hypertension or diabetes. The benefits resulting from physical exercise make the prescription in a medical context a vital method for health control and improvement of the population's quality of life. This thesis will address the development and the publication process of a mobile application on Android that tracks physical activity using GNSS. This application is designed especially for a medical context, where health professionals have the possibility of effectively prescribe a physical exercise plan that appropriately addresses the needs and limitations of the patient. This application also allows the analysis of the performance and the consequent adjustment of the plan according to the evolution of the user. This data can be shared between devices by using a QR Code based communication system. This mobile application was developed in Android Studio using Java and was based on an MVP (Model View Presenter) architecture pattern. The results revealed the practicability of the technology used for the recording of physical activity like walking or running. However, in an urban environment, there are some vulnerabilities in the data acquisition but they do not jeopardize the main goal of this application.**

I. Introduction

According to WHO [1], regular physical activity provides significant benefits to human health. Walking and running are some of the most common forms of physical activity that can offer multiple health benefits. Physical activity can act as a protective factor for cardiovascular disease, hypertension, diabetes, and some types of cancer. It is also associated with improved mental health, delayed development of dementia, and improved quality of life and well-being. According to the recommendations of DGS [2] and the U.S. Preventive Services Task Force [3], each adult should aim to accumulate 150 minutes of moderate or 75 minutes of vigorous physical activity per week.

All these benefits mentioned make the prescription of physical exercise in a clinical environment an important tool for controlling health and improving quality of life. However, there are some obstacles that hinder its success. On the one hand, there are difficulties both in transmitting information about the intended prescription and in verifying its execution. On the other hand, this type of prescription is different from a classic drug prescription. While a prescription for a medication is often a simpler and a finite-term medical indication, a prescription of a physical activity such as walking or running is usually associated with a task that consumes a significant amount of time and should be performed on a regular and long-term basis. The identification of these problems raises the need to create a solution that facilitates the doctor-patient communication and helps to consensually ensure an effective prescription and monitoring of physical activity.

The main goal of this thesis is the development of a mobile application in Android that tracks and monitors physical activity. The tracking of physical activity is intended to be done through the collection of location data that, after internally processed, generates multiple activity statistics such as distance traveled, duration of activity and average speed.

Another objective of this project is to integrate features into the application that help monitor and customize the activity to be performed by the user according to his profile and condition. Therefore, it is intended to create an interface where it is possible to customize the type of goal desired (spatial or temporal), the weekly distribution and the expected intensity of their activity. It is also intended to create a mechanism that allows sharing activity data. This functionality is intended to meet the need for a simple and effective process for sharing performance data with health professionals.

It was established as a final goal to make this application publicly available in Google Play Store.

This project has the support and collaboration of an internist with the specialty of General and Family Medicine. The main purpose of this partnership is to share the necessary knowledge in order to create a final product suitable to the needs of the clinical use context. Besides discussing the initial objectives and requirements of the application, it is intended to establish constant communication throughout all phases of the work.

II. State of Art

A. Physical Activity Mobile Applications

There are numerous physical activity monitoring mobile apps, with "Strava", "Running App" and "Map My Run" being some of the most commonly used for recording various types of physical activity by using the GNSS receiver, such as running or walking. In addition, these apps allow the access to personal challenges and share your progress on a social networking system.

However, there are some aspects that make these applications less suitable for a clinical context. Besides generally having a complex interface, the data sharing systems of these applications are not specially developed for a clinical environment. On the one hand, these functionalities are designed for a more social context, where there is a bidirectional relationship when sharing data. On the other hand, the data presentation format is poorly focused on the essential information that is needed to be shared in a medical context.

B. Navigation Systems for Mobile Devices

GNSS is a general denomination for the various satellite navigation systems that allow users with compatible receivers to determine their position and time by processing the signal sent by satellites in the network. These signals come from a number of constellations in various countries. The main ones include the GPS, Galileo, GLONASS and Beidou systems. A GNSS-supported navigation receiver like a smartphone may not be limited to the use of a specific system, so much so that it can have access to a wide range of GNSS services. This contributes to better availability, quality of service and better accuracy in estimating the position of the device. [4]

In addition to GNSS, it is also possible to obtain location information using Wi-Fi. Although less accurate, this source can be used as a complement to the satellite positioning system. The processing of these signals in Android is done using a Fused Location Provider API by Google. This API receives the signal from one or more sources and estimates the device location.

In an increasingly digital world, navigation systems have become a fundamental pillar for the normal functioning of today's society. In 2019, 6.4 billion devices were using GNSS and this number is expected to grow to 9.6 billion in 2029. One of the most usual type of user is the average citizen, where this system is mostly used as a navigation and location system. In addition, this system is also used as a tool in various sectors of society worldwide, such as agriculture, construction, transport, mining, logistics and military [4] [5].

III. Technologies

A. GNSS Systems

Each GNSS system is composed of a set of satellites whose main function is to transmit radio waves to receivers that later estimate their current position. Each of these signals sends information about the position of the satellite on the network and its sending instant. With this information, the propagation time of the message is calculated, which consequently makes it possible to calculate the distance between the two communication points. The joint information from several signals allows the estimation of the receiver's current position by triangulation, such as longitude, latitude and height in relation to the ellipsoid [6] [7] [8].

One of the factors that directly affects the quality of location data acquisition is the environment around the receiver. On the one hand, open locations without buildings nearby are ideal for GNSS signal reception. However, in urban environments with tall buildings there are two major issues: the blocking of the field of view which leads to a consequent reduction of visible satellites; and multipath interference. Under normal conditions, the signal sent by a satellite propagates approximately in a straight line to the receiver, and this information is used as an assumption for estimating the receiver's position relative to the satellite. However, when multipath interference occurs, i.e. the signal is reflected until it reaches the receiver, the propagation time increases, consequently changing the estimate of the receiver's distance from the transmitting satellite, which can lead to a greater error when estimating its position [6] [8].

These systems are typically structured into three distinct segments: the space, the control, and the user segments.

The space segment is the set of active satellites that transmit radio signals to the receiving devices. Each GNSS system has its own associated space segment consisting of its constellation. These include the GPS, GLONASS, BeiDou and Galileo systems. These satellites orbit in the middle Earth orbit zone (MEO) and are evenly distributed around the Earth in order to provide complete coverage of it [6] [9].

The control segment consists of a global network of ground infrastructure whose purpose is to monitor and correct the activity of the satellites in the network. In addition to estimating the location and the integrity of the satellites, these stations also have the ability to send and correct information from the satellites such as the internal clock setting. [6] [9].

Finally, the user segment refers to the entire universe of users who are receivers of the GNSS signal transmitted by the system's satellites. The most common users include mobile devices, cars and military devices [6] [9].

B. Coordinate Reference System

A coordinate reference system is a model that characterizes, locally or globally, a given geographic point. Depending on the usefulness there are several systems used in order to make this characterization. To understand each of these systems it is first necessary to study the complex and irregular shape of the Earth.

The geoid is a surface that roughly defines the irregular shape of the Earth. This model is based on a simpler model of representing the Earth's shape called the terrestrial ellipsoid. The ellipsoid is based on two reference parameters of the Earth's ellipsoid model: the semi-major axis (a) and the flattening coefficient of the ellipsoid (f) [10] [11].

Throughout history several ellipsoid models have been proposed with different combinations of a and f values. One such version of the ellipsoid is the WGS84 model, determined in 1984 and still used today as a reference model by the GPS system and the Google's location API. For this particular model the following reference values are [10] [11]:

$$a = 6378137.0m \qquad f = \frac{1}{298.257223563}$$

However, the irregular shape of the Earth's justifies the use of a more accurate model such as the geoid. The geoid model is a point-by-point corrected version of the Earth's ellipsoidal model, i.e. it is described by the height of its points relative to the Earth's ellipsoid, thus forming the so-called geoid curl. In Fig. 1 it is represented the hyperbolized shape of the Earth's surface based on the geoid model [11].

Coordinate systems are used to describe the position of a given point in space. Two of the most important systems for

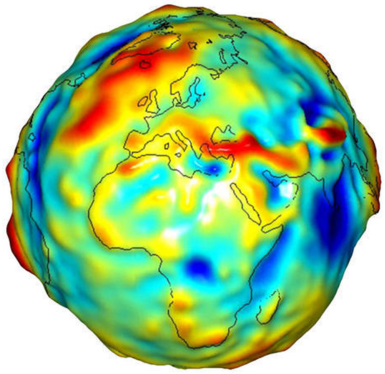

Fig. 1. Geoid model of the Earth [13]

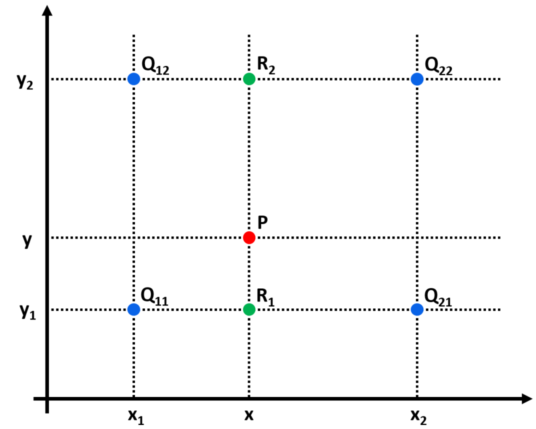

Fig. 2. Bilinear interpolation grid - adapted from [12]

characterizing a point on the Earth's surface are the geodetic coordinate system and the Cartesian system. The geodetic coordinate system characterizes a given point using three parameters: geodetic latitude (ϕ), geodetic longitude (λ) and the geodetic height (h) . On the other hand, the Cartesian Earth coordinate system is based on a three-dimensional space whose origin lies at the center of the Earth and uses the coordinates x, y and z to indicate a point in space [10] [11].

On Android smartphones, the estimation of the device's location is done using Google's Fused Location Provider API. This API receives and processes various sources of information such as satellites from the GNSS or Wi-Fi network and estimates the updated position of the device under the WGS84 referential based on geodetic coordinates. However, this coordinate system is not always the most suitable for all contexts, such as for calculating the distance between two points in space. Alternatively, the use of an alternative system such as Cartesian coordinates facilitates the calculation of this parameter. Additionally, it is also necessary to convert the ellipsoid-based data from the WGS84 format to the [10] [11].

The calculation of Cartesian coordinates (x, y, z) using geodetic coordinates (ϕ, λ, h) is given by the expressions in 1. As stated earlier, f is the tabulated value of the flattening coefficient of the ellipsoid according to the WGS84 model and R_N is the radius of curvature at the point which can be calculated as given in the expression 2 [10] [11].

$$\begin{aligned} x &= (R_N + h)\cos\phi\cos\lambda \\ y &= (R_N + h)\cos\phi\sin\lambda \\ z &= ((1-f)^2 R_N + h)\sin\phi \end{aligned} \tag{1}$$

$$R_N = \frac{a}{\sqrt{1 - f(2-f)(\sin\phi)^2)}}. \tag{2}$$

The conversion of the ellipsoid height to the geoid height is done using an approximation model called EGM. This model gathers a compilation of geoid heights relative to the ellipsoid WGS84 (G) uniformly distributed on a rectangular grid that characterizes the Earth's surface as represented in Fig. 2.

This model is available in several versions with distinct point density grids (ranges from 5', 2.5', and 1' for the EGM2008 version of the model). The G for any point on the globe is calculated using bilinear interpolation on the rectangular grid of points. Choosing the four closest points on the grid Q_{11}, Q_{12}, Q_{21} and Q_{22} and the respective G values $f(Q_{11})$, $f(Q_{12})$, $f(Q_{21})$ and $f(Q_{22})$, the following expressions are used to calculate the value of $G = f(x, y)$ for point P [12]:

$$\begin{aligned} f(x, y_1) &= \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \\ f(x, y_2) &= \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \\ G = f(x, y) &= \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2) \end{aligned} \tag{3}$$

After this, it is necessary to perform the conversion from the height relative to the ellipsoid surface to the height relative to the geoid. In Fig. 3 a representation of the three quantities involved in this calculation can be seen: the height relative to the ellipsoid (h), the height relative to the geoid (H) and the height of the geoid relative to the ellipsoid (G). H can thus be roughly determined by the following expression [10] [11]:

$$H = h - G \tag{4}$$

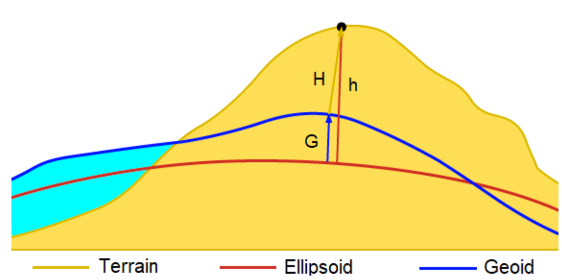

Fig. 3. Representation of ellipsoid height, geoid height, and geoid height relative to the ellipsoid - adapted from [14]

C. QR Code

QR Code is a type of barcode formed by a two-dimensional matrix. It is usually used with a camera of a mobile device and is generally intended to encode and transmit information that can later be read by other devices. This code can be used to encode simple text, URLs, personal contacts, among others. Beyond the simplest, everyday use, this technology is used in multiple sectors of the economy such as industry, logistics and transportation. [15] [16].

The storage capacity of this code depends directly on two important factors: the degree of redundancy and the maximum information density. Many times the readability of a QR Code is hindered by the low quality of the camera, the angle and the inadequate luminosity at the moment of capture. With this, the existence of this redundancy mechanism becomes important to facilitate the readability of this code. Depending on what is intended, the degree of redundancy usually varies between 7 and 30%, and the storage capacity is consequently affected. In addition, code readability is affected by the information density of the code. Above a certain density, the code becomes unreadable due to factors such as the resolution of the screen transmitting the code and the capture resolution of the camera reading the code [15] [16].

IV. Logical Implementation

A. Location Data Acquisition and Processing

One of the main functionalities to be implemented in the application is the recording and calculation of statistics of a physical activity, such as the distance traveled, average speed, altitude and time. One of the possible solutions is to periodically acquire GNSS location data. Each update contains a set of data regarding the location of the device, such as latitude, longitude, altitude, Timestamp, among other information. For each of these updates it is necessary to perform some logical processing before it is possible to calculate the desired statistics, namely the distance traveled or the height relative to the mean sea level (height relative to the geoid).

The determination of the distance travelled in an activity is done by successively calculating the distance between the points obtained by the GNSS receiver. This calculation is performed using the Euclidean method for the distance between points with Cartesian coordinates. Thus, it is necessary to convert the points provided on the geodesic coordinates format to the Cartesian format. This process is performed based on the expressions 1 and 2.

On the other hand, one of the data obtained by the GNSS receiver is the height relative to the WGS84 model ellipsoid at the receiver point. For this reason, it is necessary to perform a conversion from height relative to the ellipsoid to the height relative to the geoid. This conversion is done by reading the file EGM2008-5 which contains a compilation of geoid heights relative to the ellipsoid. The grid points are spaced 5' apart in latitude and longitude, and it is necessary to perform bilinear interpolation to calculate the value for a given point on the globe. The geoid height calculation is obtained by subtracting from the ellipsoid height (provided by the GNSS signal) the geoid height relative to the ellipsoid obtained by bilinear interpolation.

Another important aspect in this part of the application is to ensure that the collection of location data is done consistently and without interruption. In an application of this type, where location data is usually recorded for relatively long periods of time, it is vital that all this processing continues to be done even when the application is put in the background, i.e. when another application is opened or when the user locks the mobile device. The solution is to use a Foreground Service with active notifications. This approach ensures that the application, while recording an activity, is never placed in the background. This happens because there is a permanent notification in the notification bar of the device. This notification forces the application to remain active and running. On the other hand, this notification also serves as a reminder to the user that the app is running and collecting location data.

When the recording of a physical activity is finished, it is necessary to calculate and store all the general statistics in the internal memory of the device, so that they can be used in other parts of the application. To store this data it was used a MySQL database.

B. Goal Setting

Another feature implemented in the application is the mechanism where goals are set for the user's activity. This function serves as an aid tool for monitoring physical activity, where goals are set (distance or time) and can later be used to analyze the performance of the activity performed by the user.

This information is stored using a MySQL database. Each table entry stores information about the Timestamp, the type of goal set, the minimum and maximum intensities and the goal values for each of the days of the week. Changing any of these parameters generates a new table entry. The set of all table entries forms the history of all the goal data.

C. Data Communication

Another feature of this application is the activity data sharing system. This feature is mainly intended to be used in a medical context where the patient has the ability to share his activity history. With this information, the doctor's device is able to calculate general statistics and present the data in a clear and uncomplicated way, thus allowing a more effective analysis of the patient's physical activity performance. This functionality can also be applied in the fitness context, where a personal trainer can monitor his client's performance data.

This functionality was implemented based on the transmission of information using a QR Code. One of the stages of the activity data communication process is the process of sharing data by encoding and creating a QR Code that is composed of several steps. To understand each one, it is first necessary to understand the type of information needed to be transferred.

One way to condense this data starts by grouping all the activity statistics stored in daily statistics. Taking an example, two different activities are recorded in a single day: one where

2 kilometers were traveled in 20 minutes, and another where 3 kilometers were made in 40 minutes. In the data sharing process, instead of sending the information for each of these activities, the two activities are grouped into a daily statistic (in this case, the information to be sent this day is that 5 kilometers were traveled in 60 minutes).

In addition, it is necessary to ensure that all the relevant information is sent. On the one hand, it is necessary to ensure that it is sent the information about all days on which an activity was performed, including days on which there was no daily objective set. On the other hand, it is also necessary to send the information of the days on which daily goals were set, including days on which the user didnt record any activity. Together, these two groups contain enough information to characterize the entire history of user activity.

All this data is grouped into a single daily data set. Each daily dataset is called a daily package. Each of these packages will thus have the data strictly necessary to characterize each day's activity, such as distance, time, Timestamp, minimum intensity, and goal type/value.

It is from this daily package that the message to be encoded in the QR Code is designed. In Fig. 4 it is possible to schematically visualize the whole process up to the creation of QR Code.

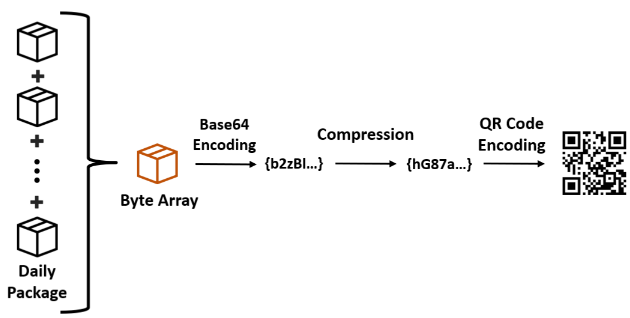

Fig. 4. QR Code encoding process

In a first step, each of the daily packets is joined in order to originate a byte vector. After this step, the Base64 encoding method is used, which converts the information contained in the byte vector into a *string*. Next a data compression function is used. The use of this procedure is appropriate in this context for two reasons: firstly, the space occupied by the message is significantly reduced; secondly, although there is extra processing in the compression and decompression of this data, the order of magnitude of the data is small enough for this extra processing time to be unnoticeable. Finally, the QR Code is encoded and then displayed on the user's device.

The process of decoding the QR Code happens similarly to the encoding process, although in reverse order. Fig. 5 is represented a schematic with the whole process of reading and interpreting the QR Code.

V. Application

The mobile application "TocáMexer" was developed in Android Studio using Java and its main functionality is to record

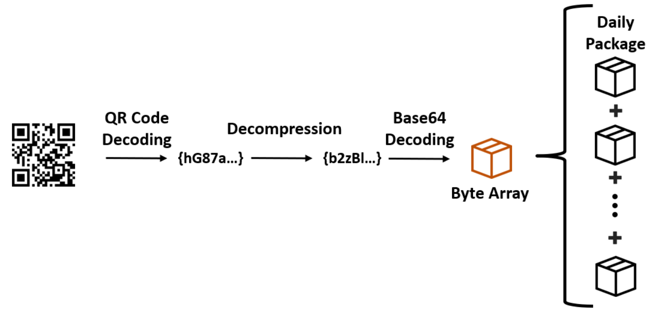

Fig. 5. QR Code decoding process

and monitor physical activity using GNSS. Each recorded activity has information about the distance traveled, time, average speed and positive elevation. In addition it is possible to set daily goals in order to compare the user's progress with the expected one. All this information is used to generate reports about the user's performance that can optionally be shared with other devices using a QR Code based system.

A. Home Page

The application starts with the Home Page Fragment, as shown in Fig. 6 and 7.

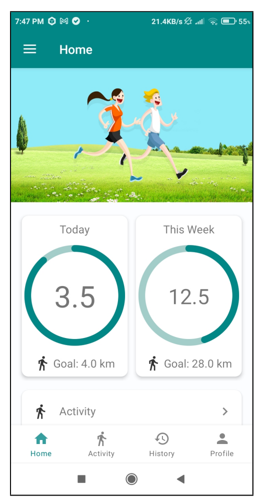

Fig. 6. Home Page top interface

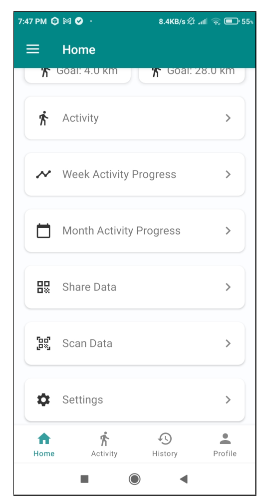

Fig. 7. Home Page bottom interface

At the top of the screen two Cardviews are displayed with general information about the user's activity. Depending on the the type of goal chosen, a circular graph is displayed regarding the current progress in comparison with the predetermined goal. The Cardview on the left side shows information about the next daily goal to be accomplished and its progress, while the Cardview on the right side shows information about the weekly progress compared to the respective expected goal.

Below this element it can be find a shortcut button to access the Activity Fragment. Below this are two rectangular buttons that allow access to more detailed information about the user's physical activity progress. Pressing the "Weekly Progress" button displays a mixed graph of dots and bars (picture 8) where it is possible to compare weekly progress with the daily goal previously set. By pressing the ”Monthly Progress” button, a color-coded monthly calendar is displayed (Fig. 9) where it is possible to have a more general perspective on the progress compared to the goals set by the user throughout a month. In this calendar view a color code is displayed that evaluates the performance of each day compared to the goals set. In this case, each green dot identifies a day when the daily goal and the minimum intensity goal were both achieved. On the other hand, orange identifies days when the daily target was met although without meeting the set minimum intensity. Red are the days on which the daily goal was not achieved. Finally, in purple are identified the days on which an extra activity was performed on a day with no daily goal established.

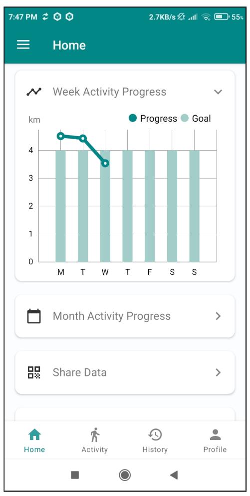

Fig. 8. Weekly Progress interface

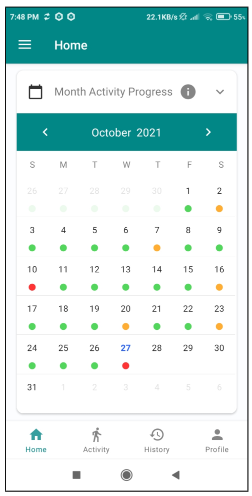

Fig. 9. Monthly Progress interface

B. Activity

The main functionality of this Fragment is to record an activity by periodically collecting and interpreting location data. By pressing the green ”Start Activity” button, the interface updates to show the ”Pause” and ”Stop” buttons, as represented in Fig. 10. Furthermore, this action triggers the start of the location data acquisition obtained by the Google Location API. It is with this information that parameters such as distance traveled over time, speed, and height relative to mean sea level are successively calculated and updated. Together with the time elapsed since the start of the activity, all these parameters are continuously updated in the upper part of the screen of this Fragment.

Below these values it can also be find two visual animations with information about the current state of the activity.

The first animation with the name “Intensity” lets the user know if he/she is performing his/her physical activity within the intensity pre-determined in the Profile Fragment settings. This graph presents three colored zones: blue represents the set of speeds below the set minimum pace; red gathers the set of speeds above the set maximum pace; green comprises the set of speeds above the minimum pace and below the predetermined maximum pace. The current speed is represented by the largest vertical bar in the graph.

The second and last graphical representation named ”Daily goal progress” gives information about the user’s progress according to the goal set previously for the day of the activity. This information is updated throughout the activity and can represent one of two types of daily goal: distance in kilometers (km) or time in minutes (min). If the user does not set any goal, this graph is not visible.

Finally, it is possible to access the spatial representation of the journey travelled by clicking on the button with the map icon in the upper right corner of the application. With this action the application changes its interface and starts showing the real time path taken along the route using a Google Maps map (Fig. 11). The interface from Fig. 10 can be shown again by clicking the back icon in the upper right corner of the application.

The activity ends by pressing the red ”Stop” button. With this action, all previously mentioned interface components (with the exception of the goal progress graph) return to the initial state.

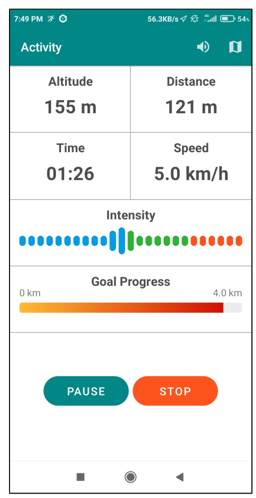

Fig. 10. Interface with an ongoing activity

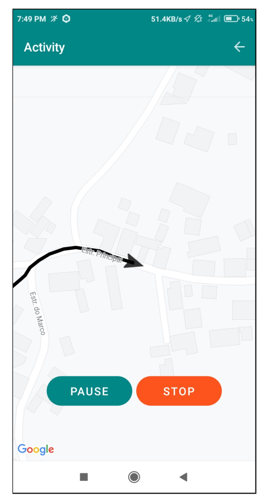

Fig. 11. Activity map interface

C. History

Similarly to the Main Page, in this Fragment two graphical contents are presented: a graph where it is possible to compare the weekly progress with the daily goal previously established; and a calendar with a monthly view with information about the user’s performance. These elements behave identically to the graphics described in the Home Page section. By clicking on the icon in the upper right corner, the Storage interface appears. This interface displays the history of activities performed in chronological order. For each activity information is presented about the activity start timestamp, the time, the distance, the positive altitude variation and the average speed. Each of these activities can be deleted by dragging Cardview horizontally (either to the right or to the left).

D. Profile

In this Fragment (Fig. 12) it is possible to customize several aspects of the user's goal monitoring feature. At the top of the screen is the "Goal Type" element where it is possible to select either time or distance as the reference for monitoring the user's activity. Below this is the "Intensity" element, where it is possible to customize the minimum and maximum speeds expected for the user's exercise. Finally, there is the "Weekly Distribution" element, where it is possible to choose the goal value for each of the days of the week.

E. Share Data

This Fragment (Fig. 13) aims to share the user's data with other devices that also have the same app. The shared data concerns the user's most recent physical activity history. This feature generates a QR Code with all the information encoded to be transmitted to the target device.

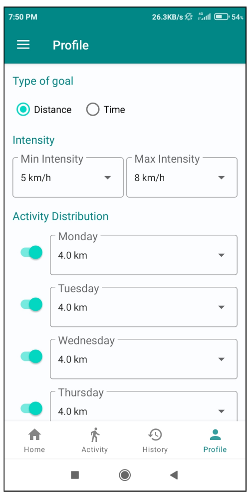

Fig. 12. Profile interface

Fig. 13. Share Data interface

F. Scan Data

This Fragment serves as a complement to the previous Share Data Fragment. This feature is used on the device where the user wants to get the data from another device. The scan is performed by pointing the camera at another device that has the data sharing feature open.

After decoding the information, the interface changes to a statistical data display interface similar as shown in Fig. 14. At the top of the screen it is displayed two Cardviews with the general information about the overall progress of the user. Below these elements is a monthly calendar where it is provided an overview of the daily progress of the user being analyzed. The color coding used is entirely similar to that referenced in the "Monthly Progress" element in the Home Page segment.

G. Settings

In this Fragment (Fig. 15) it is possible to customize several settings about the general operation of the application.

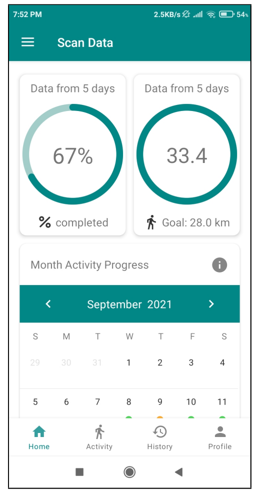

Fig. 14. Scan Data interface - data report

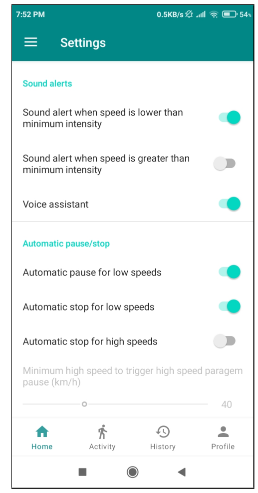

Fig. 15. Settings interface

At the top of the screen it is displayed the "Sound Alerts" section where it is possible to manage various types of sound alerts available in the application. The top two work on the basis of comparing the user's speed with the established minimum and maximum speeds of the Profile Fragment. If, for example, the first alert is active and the user does some part of his activity below the set minimum intensity, a sound will be played informing him of what has happened. The second maximum intensity alert works in a similar way when the user's speed exceeds the set maximum intensity. Finally, there is the "Voice Assistant". This feature uses a "Text-to-Speech" API that interacts with the user at various points in the activity, such as indicating when the daily goal is met, for example.

Further down the screen it is displayed the "Auto Pause" section. In this part of the screen it is possible to enable features such as the automatic pause for low speeds, the automatic stop for low speeds, and the automatic stop for high speeds. As its name indicates, activating the first one allows the application to detect when the user temporarily stops in some part of his activity and pause its recording. The activity starts counting again as soon as the user resumes his movement. This feature can be useful if it is intended to only record useful physical activity, avoiding tracking moments when the user is momentarily stopped. The automatic stop works in a similar way to the automatic pause, with the particularity of permanently ending the activity when it is detected that the user has not moved for a long period of time. Finally, activating the automatic stop for high speeds allows the device to definitely end an activity when a higher-than-expected speed is registered. This kind of situation can occur when the user forgets to stop tracking an activity and then uses a transport vehicle. The speed at which this system is triggered can be customized according to the user's preferences.

VI. System Tests

A. Data Acquisition Interval Test

The main purpose of this test is to validate the quality of representation of a given path. The tests were performed on a circuit with approximately 511 meters near Batalha. This site is located in a rural environment near low-rise housing. Furthermore the chosen circuit is characterized by containing curves and straights, an environment that is close to most of the contexts in which this application will be used. This path was traveled several times to collect location data for different acquisition time intervals. Furthermore, tests with different characteristics were performed: one at walking pace and the other at running pace.

Figures 16, 17, 18 and 19 represent the trajectories obtained in walking pace in four tests with different data acquisition intervals: 1, 2, 4 and 8 seconds, respectively. In addition, in table I it is possible to verify some results associated with the four tests performed in this environment.

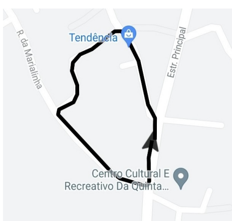

Fig. 16. Test with walking pace with Δt = 1s

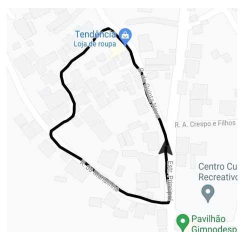

Fig. 17. Test with walking pace with Δt = 2s

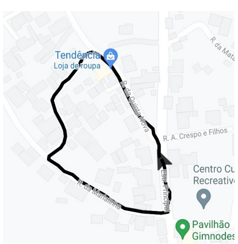

Fig. 18. Test with walking pace with Δt = 4s

Fig. 19. Test with walking pace with Δt = 8s

TABLE I
Information on the tests performed at walking pace

Time interval (s)	1	2	4	8
Average speed (km/h)	6,0	6,0	5,6	5,7
Measured distance (m)	512,1	507,4	506,8	481,0
Distance relative error (%)	1,9	2,8	2,9	7,9

Figures 20, 21, 22 and 23 represent the trajectories obtained at running pace, in four tests with different data acquisition intervals: 1, 2, 4 and 8 seconds, respectively. In addition, in table II it is possible to verify some results associated with the four tests performed in this environment.

It is possible to conclude that, tendentially, the larger the interval of acquisition of location data, the worse is the quality

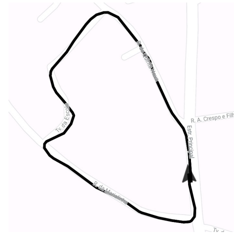

Fig. 20. Test with running pace with Δt = 1s

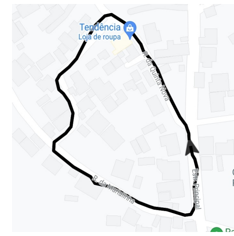

Fig. 21. Test with running pace with Δt = 2s

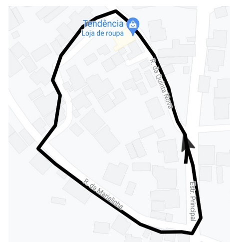

Fig. 22. Test with running pace with Δt = 4s

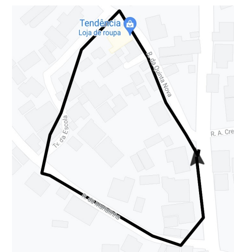

Fig. 23. Test with running pace with Δt = 8s

of representation of a given path on the map. In comparison with the other intervals tested, for Δt = 8 s a high percentage of points were mismatched with the expected trajectory, particularly in curves. It is also possible to conclude that, tendentially, the choice of the acquisition interval affects the accuracy of the distance traveled calculation. The acquisition performed with an interval of 1 second was the test where the smallest distance error was obtained (between 0.5 and 1.9 %), whereas for Δt = 8 s a larger error was obtained (between 7.9 and 11.2 %). Furthermore, it is possible to conclude that the rhythm of the activity interferes directly with the quality of the capture of the performed path. On the one hand, the tracks performed at walking pace tended to present the best results. On the other hand, the tracks performed at a more intense pace tended to be further away from the expected result (511 meters).

In the practical context of this application, these results suggest that the most appropriate data acquisition interval is 1 second. However, the choice of the this interval could give rise to some reflection if the implementation of the application was based on the real time delivery over the Internet. The power consumption in an application with these characteristics would increase with the increase of the data sending frequency. In this case, the most appropriate time interval to minimize energy consumption without compromising the quality required to obtain data could be between 2 and 4 seconds.

TABLE II
Information on the tests performed at walking pace

Time interval (s)	1	2	4	8
Average speed (km/h)	10,5	10,0	11,2	10,9
Measured distance (m)	524,6	507,6	498,3	463,7
Distance relative error (%)	0,5	2,8	4,6	11,2

B. Precision Test

This test aims to compare the accuracy in obtaining location data in a rural and urban environment. For the rural environment the same location was chosen as in the previous test. On the other hand, the urban environment selected is located in the center of the city of Leiria. The route traveled is characterized by containing medium height buildings and narrower streets, compared to the route in the rural environment.

For each of the locations, tests were performed in order to obtain 500 acquisitions of location data. The purpose of these two tests is to compare the quality of data reception using the accuracy parameter calculated in each of the location data packages. This parameter is obtained from Google's location API by estimating the radial error of the location in meters with a confidence interval of 68 %. Figures 24 and 25 show two distribution histograms of the accuracy values in meters obtained during the test in rural and urban environments, respectively.

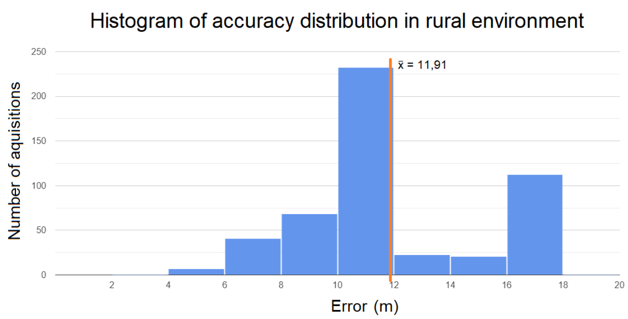

Fig. 24. Histogram of accuracy distribution in rural environment

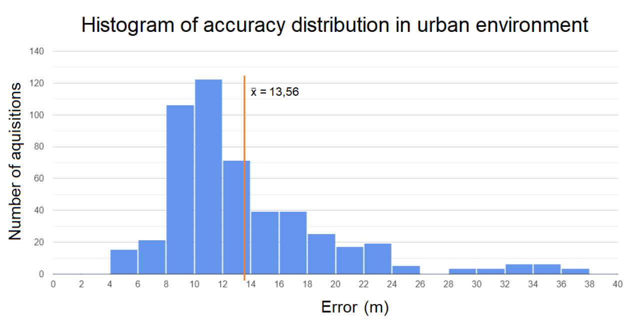

Fig. 25. Histogram of accuracy distribution in urban environment

The results obtained allow us to conclude that there is indeed a difference in the accuracy of the data obtained in a rural environment compared to an urban environment. On average, the accuracy calculated in a rural environment had a value of 11.91 meters, while for the urban environment an average of 13.56 meters was obtained. Although the difference between the averages is reduced, in some of the points obtained in an urban environment a higher error was obtained (up to 38 meters), a situation that did not occur in a rural environment. This difference can be explained by the narrow morphology of the streets and the presence of tall buildings.

VII. App Publication

The procedure for publishing "TocáMexer" application began on September 2th with the completion of an initial registration form on Google Console. This form provided information such as the name of the application, the official language and the type of application. After this step, it was provided access to a control panel where multiple settings related to the testing and launching of the application could be managed.

In a next step, it was started the distribution of the application as an internal test. The main purpose of this type of test was to easily perform a quick distribution of a certain initial version of the application to a small group of people. The first distribution of this version occurred on September 4, and 5 different versions were distributed over a period of 6 days.

Afterwards, a closed test version was released to test the application on a wider set of people. This process began on September 11th where a variety of versions were released over a 24 day period. Unlike the internal test version, this required the completion of additional information about the application and the configuration of the app store page. This information was provided in order to ensure that the application is secure for the intended users. Firstly, a privacy policy document had to be submitted to share how confidential user data is handled and whether or not the application contains advertisements. On the other hand, details about the type of application and the target audience were provided. In addition it was necessary to describe how any confidential permissions are used within the application. After submitting this information, the application was subjected to a Google Play Store review process. In addition to the review of the aforementioned information, four types of robotic tests were performed:

- Stability test - checks if the application crashes or stops responding during application execution;
- Performance test - identifies problems with startup time and slow rendering;
- Accessibility test - checks the ease of use of the application in terms of user interface;
- Security and trustworthiness test - identifies security vulnerabilities that may pose a risk to the users;

At a later stage, on October 7th an open test version was released in order to make the application publicly available on the Google Play Store. At this stage of the process, anyone was able to participate in this test program and try out the application. In Fig. 26 it is shown the Google Play Store Page for this version of the application.

VIII. Conclusion

The main goal of this work was to develop and publish an Android application that tracks physical activity using GNSS.

In an initial phase, the relevant specifications were defined in collaboration with an internist in Family Medicine Doctor. This collaboration proved to be very important as it allowed to clearly establish the essential requirements for the creation of a product suitable to the needs of a clinical context. Another

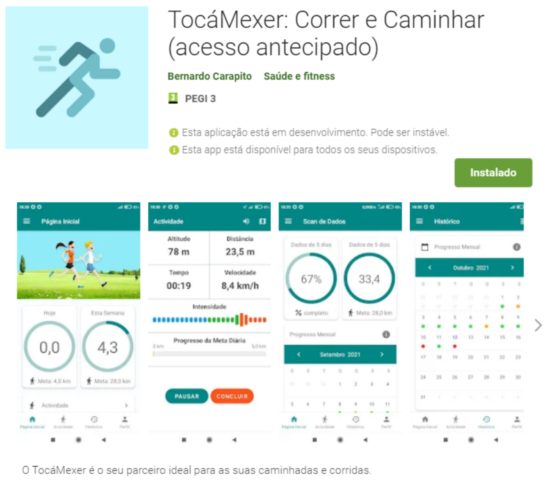

Fig. 26. Application store page

crucial aspect in the development of the application was the constant contact with other people who, over time, were testing the application. This interaction allowed collecting feedback related to the various functionalities implemented over time, thus allowing the correction of problems such as interface errors, bugs and implementation flaws. This interaction also allowed the identification and correction of particular errors that arose for various smartphone models, such as layout errors (due to the variety of screen resolutions) or errors related to the differences between operating system versions.

The results obtained in chapter VI were important to demonstrate the viability of this application, especially in a non-urbanized environment. However, this solution presented some flaws when it comes to obtaining location data in a urban environment. One of the possible explanations is the existence of tall buildings that contribute to blocking the field of view for the satellites and to the occurrence of multipath interference.

Despite these vulnerabilities encountered, it is possible to conclude that all the goals initially set were successfully met. Besides the complete development of all the intended functionalities, the goal of publishing the application in the Google Play Store was also achieved.

For future work, there are numerous aspects to be reviewed. One of the limitations of the application is the size limit for sharing information using a QR Code. One possible solution would be the development of a system where multiple QR Codes are consecutively sent and scanned until all the necessary information is transmitted. Another aspect that could be improved would be the implementation of a two-way data communication feature. In addition to the already implemented QR Code performance data sharing, this functionality could be expanded to allow the physician to transmit the prescription to the patient without the need to manually enter the goals.

One way to solve the problem of data collection in highly urbanized areas could be the development of a more complex location data processing system. Knowing that streets and roads are one of the main usage environments for this application, one way to make the trajectory calculation smoother would be to use Google's Roads API, namely the "Snap to Roads" function. This service returns the best match of the geometry of a road trajectory to a given set of coordinates sent by the device. However, there are some drawbacks to this solution. First, in urban environments, where the error is sometimes high, the path matching may not be accurate due to the high density of streets in such environments. On the other hand, there are costs associated with the use of this feature.

Finally, the realization of this project was a great personal challenge. Besides deepening my knowledge about GNSS and telecommunications in general, this work allowed me to have my first experience both in the development of a large mobile application and in its publishing process in Google Play Store. Despite many difficulties I had during the development of this project, I feel that this application can be an asset for use in a medical context.

References

[1] World Heath Organization. Physical activity. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/physical-activity

[2] Direção Geral de Saúde. Relatório Sobre Atividade Física 2020, SNS e DGS. [Online]. Available: https://noticias.ecosaude.pt/2021/04/27/relatorio-sobre-atividade-fisica-2020-sns-e-dgs/

[3] U.S. Preventive Services Task Force. Healthy Diet and Physical Activity for Cardiovascular Disease Prevention in Adults With Cardiovascular Risk Factors: Behavioral Counseling Interventions. [Online]. Available: https://www.uspreventiveservicestaskforce.org/uspstf/recommendation/healthy-diet-and-physical-activity-counseling-adults-with-high-risk-of-cvd

[4] European Global Navigation Satellite Systems Agency. GNSS Market Report Issue 6, October 2019. [Online]. Available:https://www.euspa.europa.eu/gnss-market-report-issue-6-october-2019

[5] E. Kaplan and C. Hegarty, Understanding GPS: Principles and Applications. Artech House, 2006.

[6] G. P. Petropopoulos and P. K. Srivastava, GPS and GNSS Technology in Geosciences, ElsevierScience Publishing Co Inc, 2021

[7] European Union Agency for the Space Programme. What is GNSS?. [Online]. Available: https://www.euspa.europa.eu/european-space/eu-space-programme/what-gnss

[8] H. L. B. Hofmann-Wellenhof and E. Wasle, GNSS - Global Navigation Satellite Systems: GPS, GLONASS, Galileo & more, Wien: Springer, 2008.

[9] Navstar Global Positioning System. Navstar GPS Space Segment/Navigation User Segment Interfaces. [Online]. Available: https://www.gps.gov/technical/icwg/IS-GPS-200D.pdf

[10] Z. Lu, Y. Qu, and S. Qiao, Geodesy - Introduction to Geodetic Datum and Geodetic Systems. Springer-Verlag Berlin Heidelberg, 2014

[11] S. T. Kouzeleas, "Conversion of gps data to cartesian coordinates via an application development adapted to a cad modelling system," July 2007. [Online]. Available:http://papers.cumincad.org/data/works/att/a38d.content.03652.pdf

[12] M. R. Khosravi and S. Samadi, "A blind scalable edge-guided reconstruction filter for smart envi-ronmental monitoring through green iomt-uav network, "IEEE Transactions on Green Communicationsand Networking, vol. 5, pp. 726 – 736, June 2021

[13] S. S. Mohammad R. Khosravi, "A blind scalable edge-guided reconstruction filter for smart environ-mental monitoring through green iomt-uav network," June 2021.

[14] Direção Geral do Território. Modelo do Geoide. [Online]. Available: https://www.dgterritorio.gov.pt/geodesia/modelo-geoide

[15] D. C. Hiroko Kato, Keng T. Tan, Barcodes for Mobile Devices. Cambridge University Press, 2010.

[16] D. ADC, QR Code Essentials, Denso Automatic Data Capture, 2012.