

## APPS FOR MOBILE DEVICES

<b>THEME</b>	Navigating through screens
<b>FORMAT</b>	Individual, pairs or small groups
<b>PREPARATION TIME</b>	1 hours
<b>ACTIVITY LENGTH</b>	1h30
<b>DIFFICULTY LEVEL</b>	Easy

### PEDAGOGICAL GOALS

- Continue to familiarize yourself with the AppInventor environment
- Work on implementing a new application
- Know the functionality of transmitting values from one screen to another and storing values

### NECESSARY MATERIALS

Computer room with computers connected to the Internet

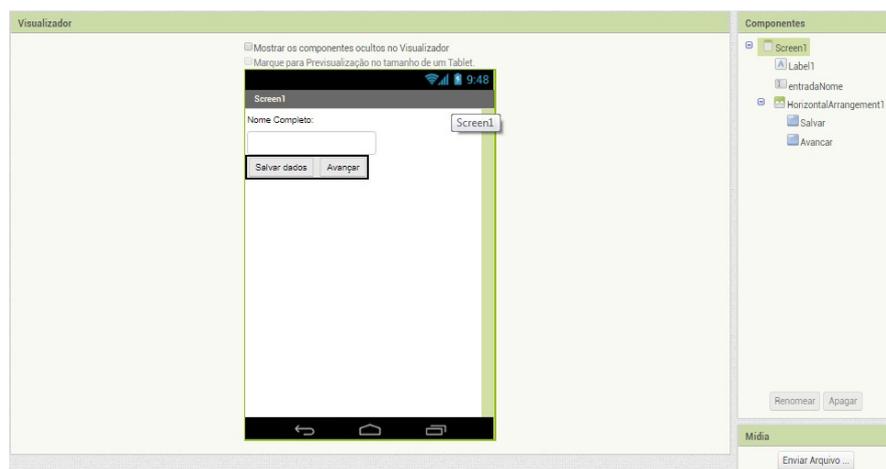
## APPS FOR MOBILE DEVICES

### Preparation:

- Ensure that the computers used have internet access.
- Study the application's interface and programming implementation through the Code IoT platform.

### Conducting the activity:

- Divide the class into pairs or trios depending on the number of computers available at your school.
- Ask the class what is data, asking for examples. Supplement with the necessary and correct information.
- Ask students what types of data we could store and transmit from screen to screen in an application, writing ideas on the board.
- Ask students to access the App Inventor website at the link <http://appinventor.mit.edu/explore/>
- Ask them to click on “Create apps”, then on “Start New Project”, naming the project “Data Transmission – Student Name”.
- Guide students on which interface components they will need to implement the app:
  - A text label component “Complete Name”;
  - A text box “entryName”;
  - Two buttons: one to “Save” the data and other “Next” that goes to the next screen.
- Guide students on how to arrange components on the screen:



- Instruct about the database “TinyDB” component and its implementation in the app. If necessary, this guidance and information is on our Code IoT platform.
- Guide students about creating a new screen, in the option "**Add screen...**"
- Instruct students about the components of the second screen and how to adjust them:
  - a label with the text "Stored data:"
  - a label that is empty and will display the value the user types in the previous screen
  - a "Back" button that, when clicked, will return to the previous screen
  - and the “TinyDB” database

## APPS FOR MOBILE DEVICES

- Now, ask students to enter the programming tab by clicking on “Blocks”.
- Teach students the functions of the different blocks needed to implement each screen in the app and guide them in assembling these blocks.
- Now ask the students to test the app with the emulator, the aiStarter program. Keep its window open.
- As students have already done this process in the previous class, encourage them to do it without guidance.
- Ask students who are able to do it to help those who can't.

### Discussion and reflection:

- After completing the activity, discuss with your students the concepts covered in this class. See examples of some questions that can be used to start the discussion.
- What other data could we store in an application?
- Could we create an app with three screens? And four?
- Would it be possible to transmit the information between them?
- What was the hardest part of the activity?
- And the easiest one?

### Tip and extra activity:

- If there is time, an interesting activity that can be done before developing the app is the game 'Broken telephone'. In this case, the students would have to pass a 'data' to each other. At the end, ask the students what the difference is between 'transmitting data' via the broken telephone and transmitting data via an app.
- If you like, ask students to share the apps with you. In this way, it is possible to verify which students and/or groups had more difficulties and in which part of the development these difficulties occurred.
- If time permits, ask students to try to develop an application to pass other data. Students can choose what to convey, and this activity can be submitted as homework. The goal is not for them to create a perfect application, but for them to explore the possibilities that have been learned so far, use creativity, and that you can, through these activities, diagnose the reception and understanding of the class regarding the contents.

### Credits:

Ohanna Jade do Amaral (LSITec/USP)  
Irene Ficheman (LSITec/USP)