Proyecto Festival ACME

Narváez, Daniela., Segura, Camilo.
Facultad de Ingeniería Electrónica y Telecomunicaciones
Electiva- Sistemas Ubicuos
Universidad del Cauca

INTRODUCCIÓN

Los dispositivos móviles y sus aplicaciones representan hoy en día una respuesta a la necesidad de los individuos de estar en continua conexión con la información y las comunicaciones, facilitando en cualquier momento la resolución de una tarea determinada o ayudando en operaciones y gestiones del día a día (compras, pagos, entradas, entre otras).

Para el diseño de aplicaciones móviles existen ambientes de programación según el sistema operativo incluido en el dispositivo. En la presente practica se usará Android que es implementado por muchos teléfonos móviles actuales o Smartphone. En Android se puede abordar el diseño de la aplicación con lenguajes como java o XML que pueden ser intimidantes, pero también hay entornos más sencillos como App Inventor ideado por Google y retomado por el MIT (ahora llamado MIT App Inventor), que hace uso de programación gráfica (bloques) para facilitar el desarrollo, sin necesidad de tener conocimientos extensos sobre sintaxis de programación. Cabe aclarar que las aplicaciones pueden incluir tecnologías que permitan cumplir con las peticiones de los usuarios o empresas, en el presente caso se trabajó con NFC (Near Field Communication), que es una tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que posibilita el intercambio de datos entre dispositivos.

Con base en estas herramientas mencionadas se diseñó una aplicación móvil para la empresa ACME en la ciudad de Popayán en apoyo a un festival para facilitar la entrada y el pago de los asistentes con manillas basada en tecnología NFC.

2. DESARROLLO

a. Requerimientos:

La empresa ACME quiere preparar un festival en la ciudad de Popayán. Para lo que ha solicitado una aplicación móvil basada en NFC (llamada Festival ACME) que le permita:

- Entrada al festival rápido y con orden. Con las pulseras NFC los asistentes pueden entrar en el recinto solo enseñando las pulseras al personal de seguridad, el cual con la aplicación “Festival ACME” puede averiguar si la pulsera es de un asistente o no lo es (muchas personas intentan colarse en dichos festivales) o si cancelo previamente la entrada al festival.

- Pago rápido y eficaz. Los asistentes deben cargar sus pulseras con anticipación en sitios autorizados por medio de la aplicación “Festival ACME” para poder pagar su entrada y su consumo en el festival. De esta forma la empresa pretende decir Adiós a las colas para pagar, adiós a esas barras que se quedaban sin cambio, adiós a esos problemas con el datafono que no permitían cobrar a ciertos asistentes, adiós a las cajas que no cuadraban al final del festival.

b. Procedimiento:

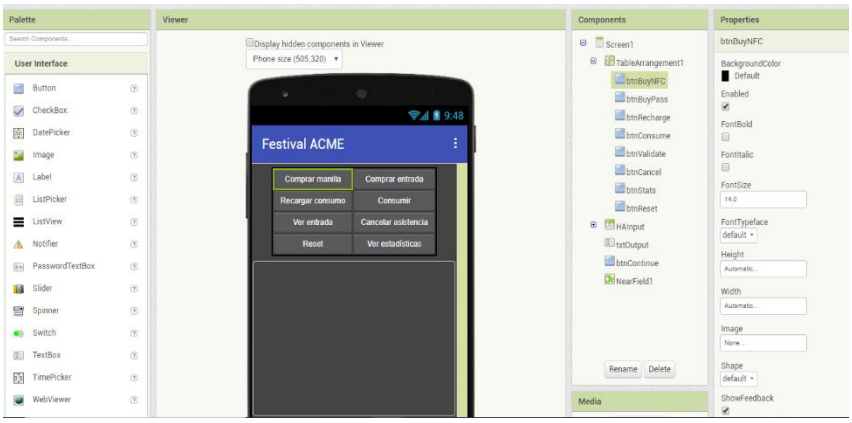
Para cumplir con los requisitos y posibles ambigüedades, se ha decidido que la aplicación debe tener siete funcionalidades principales, patentes en su interfaz principal:

1. Comprar manilla NFC. Hace la manilla NFC válida para futuros festivales ACME.
2. Comprar entrada. Hace la manilla NFC válida para el festival ACME 2020 "Endgame".
3. Recargar consumo. Recarga un valor de consumo (dinero) en la manilla.
4. Consumir. Disminuye un valor de consumo (dinero) de la manilla.
5. Validar manilla. El staff puede validar para permitir la entrada del asistente.
6. Cancelar asistencia. La persona puede acercarse a un punto autorizado para pedir una devolución y no asistir al evento. Tanto la manilla como el consumo no tienen devolución, solo el pase al evento.
7. Ver estadísticas. El staff puede ver cuantas manillas ha vendido, cuantos pases ha vendido, cuantos asistentes, cuanto se ha recargado y cuanto consumido.

Para facilitar el entorno de pruebas se decide adicionar un botón (Reset), que permite inicializar todas las variables.

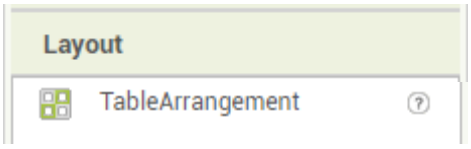
c. Diseño de interfaz:

Con las funcionalidades claras, el primer paso es construir una interfaz con dichos 7 botones así:

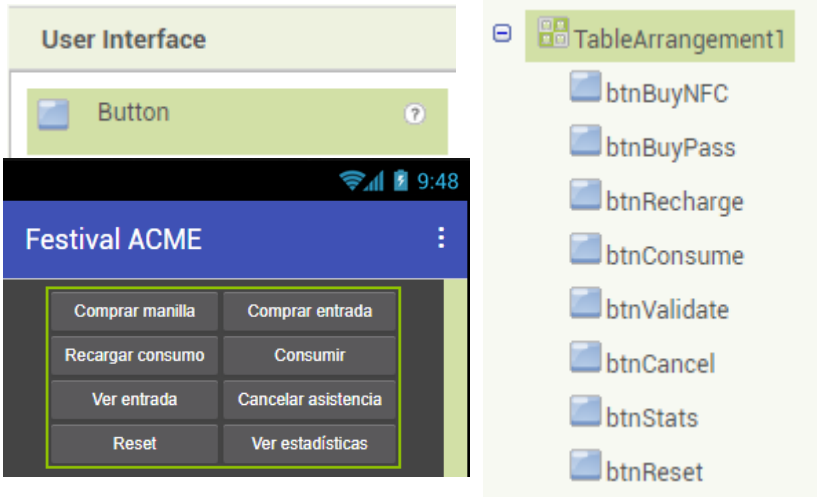


Componentes a utilizar

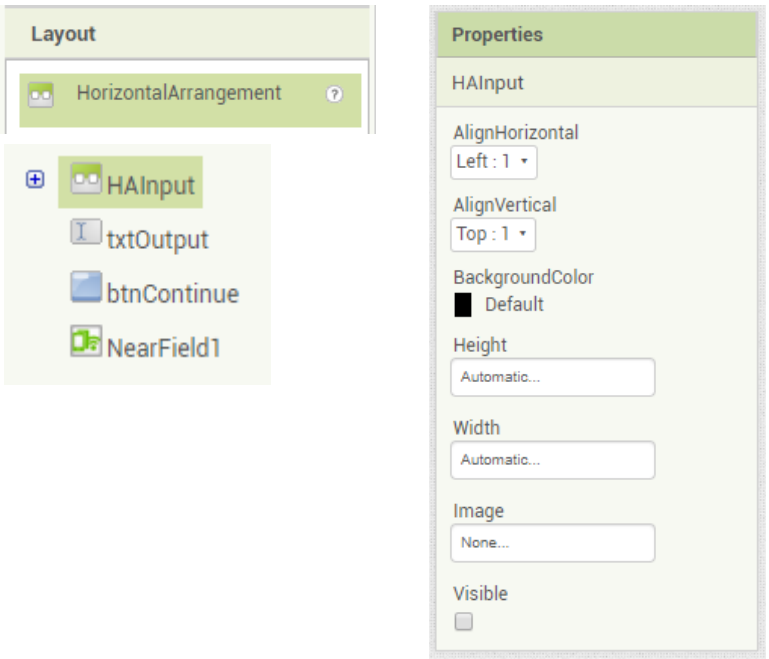
1. Screen 1
2. TableArrangement: Capa de distribución para definir el formato, y dentro del cual hay que ubicar los componentes que se desea que se representen en formato tabular



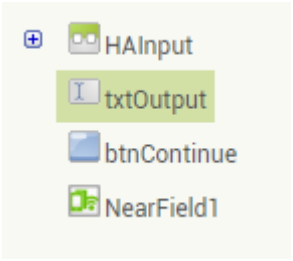
3. Button: componente que detecta cuando el usuario hace clic sobre el para realizar alguna acción en la aplicación. Estos serán ubicados en la capa de disposición anterior. Cabe aclarar que son siete botones. Su nombre y organización se verán a continuación



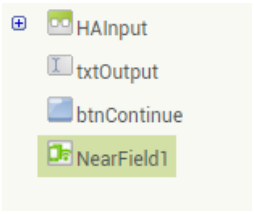
4. HorizontalArrangement: capa de disposición usada para añadir el campo de texto, componente para usar la funcionalidad NFC y un botón adicional (para confirmar procesos específicos). Esto se realiza después de la disposición de los siete botones anteriores. Además, es un elemento que se configura como no visible, desactivando la opción Visible en las propiedades del componente como se ve a continuación.



5. TextBox: componente que permite al usuario introducir y procesar texto. Se usa, ya que cada botón escribirá un mensaje en el campo de estado de abajo. Este componente se incluye en la capa de disposición anterior, y dentro de sus propiedades se coloca como visible, MultiLine y ReadOnly para el componente de salida de texto; y como no visible y NumbersOnly para el de entrada.



6. NearField: componente no visible que permite utilizar la funcionalidad NFC (leer y escribir texto en las etiquetas, solo si el dispositivo está preparado para ello). Es adicionado en la capa disposición inferior, con el fin de lograr que la manilla emita o reciba información.

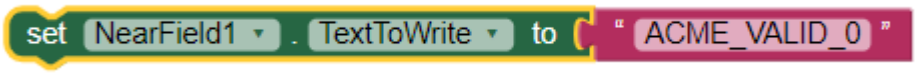


3. DESARROLLANDO LA LOGICA:

Una vez diseñada la interfaz pasamos a programar la aplicación mediante la organización de bloques de los elementos usados.

a. Criterios importantes:

Cada manilla comprada lleva un formato en texto: ACME_VALID_0

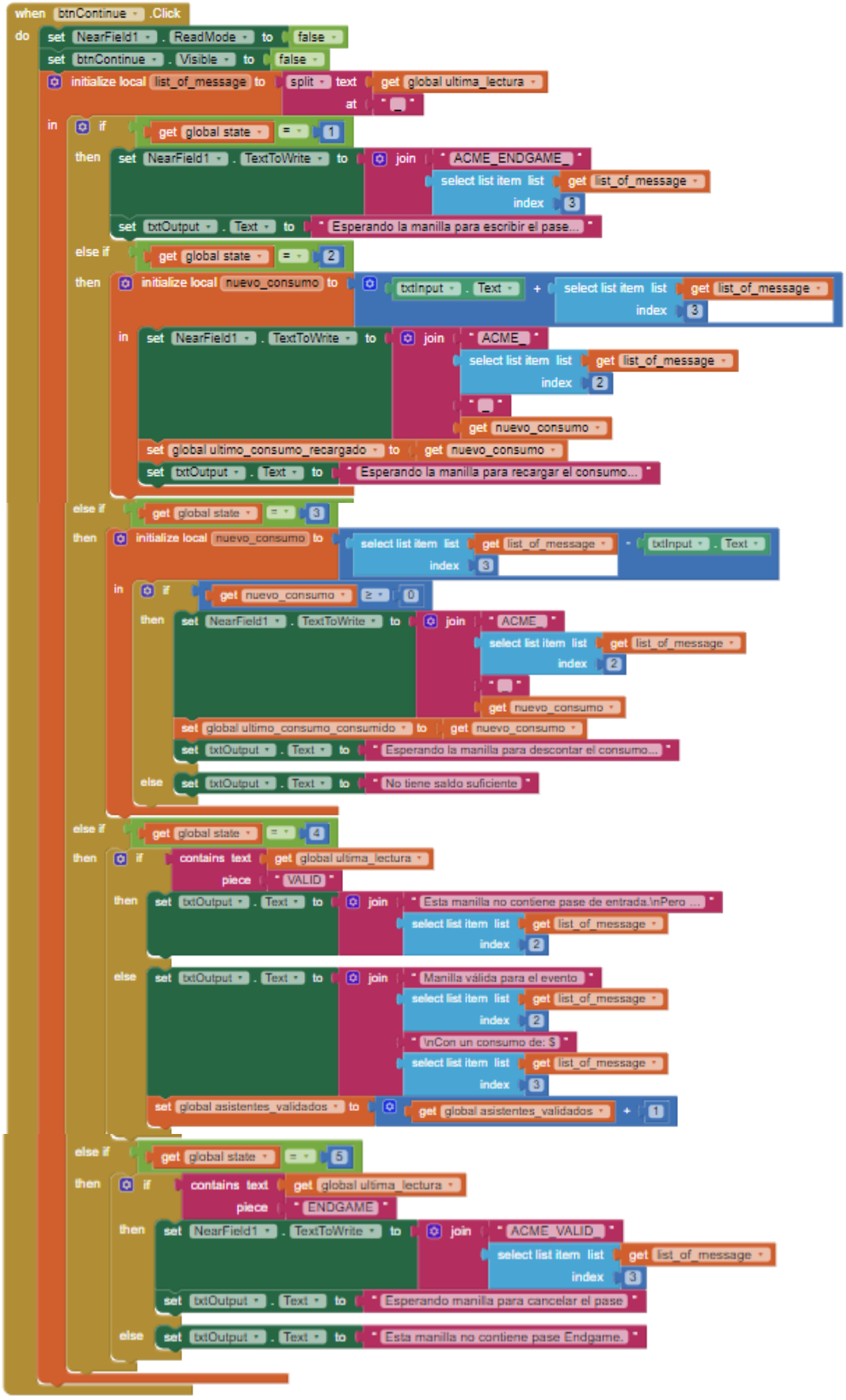


Dónde:

- "ACME" es una manilla ACME recién adquirida y válida (func 1- explicada más adelante).
- "VALID" es el estado, puede ser VALID, ENDGAME. Siendo ENDGAME una manilla válida para el evento actual del Festival ACME 2020 (func 2- explicada más adelante).
- "0" es el valor en dinero para el consumo.
- "_" es el carácter para separar

Por otra parte, se tienen las siguientes variables que serán escritas o leídas en los bloques del btnContinue y NearField1 (del componente que permite usar la funcionalidad NFC)

Dentro de la aplicación algunos botones como: comprar entrada, recargar consumo, consumir, ver entrada, cancelar asistencia, necesitan de un clic adicional para autorizar estas acciones que fue definido como btnContinue. Es decir que el desarrollo o lógica del bloque está basado en funcionalidades según el botón seleccionado, así que lo que se realiza inicialmente es la identificación del evento que se está corriendo o ejecutando en este bloque lo que se hace es asignar una función


when btnContinue .Click
do set NearField1 . ReadMode to false
set btnContinue . Visible to false
initialize local list_of_message to split text get global ultima_lectura at "_"
in if get global state = 1
then set NearField1 . TextToWrite to join "ACME_ENDGAME_" select list item list get list_of_message index 3
set txtOutput . Text to "Esperando la manilla para escribir el pase..."
else if get global state = 2
then initialize local nuevo_consumo to txtInput . Text + select list item list get list_of_message index 3
in set NearField1 . TextToWrite to join "ACME_" select list item list get list_of_message index 2 "_" get nuevo_consumo
set global ultimo_consumo_recargado to get nuevo_consumo
set txtOutput . Text to "Esperando la manilla para recargar el consumo..."
else if get global state = 3
then initialize local nuevo_consumo to select list item list get list_of_message index 3 - txtInput . Text
in if get nuevo_consumo ≥ 0
then set NearField1 . TextToWrite to join "ACME_" select list item list get list_of_message index 2 "_" get nuevo_consumo
set global ultimo_consumo_consumido to get nuevo_consumo
set txtOutput . Text to "Esperando la manilla para descontar el consumo..."
else set txtOutput . Text to "No tiene saldo suficiente"
else if get global state = 4
then if contains text get global ultima_lectura piece "VALID"
then set txtOutput . Text to join "Esta manilla no contiene pase de entrada.\nPero ..." select list item list get list_of_message index 2
else set txtOutput . Text to join "Manilla válida para el evento" select list item list get list_of_message index 2 "\nCon un consumo de: $" select list item list get list_of_message index 3
set global asistentes_validados to get global asistentes_validados + 1
else if get global state = 5
then if contains text get global ultima_lectura piece "ENDGAME"
then set NearField1 . TextToWrite to join "ACME_VALID_" select list item list get list_of_message index 3
set txtOutput . Text to "Esperando manilla para cancelar el pase"
else set txtOutput . Text to "Esta manilla no contiene pase Endgame."

Ahora, dependiendo del boton presionado (definido por el valor de global state), tambien existe un componente no visible NearField que lo traducirá en escritura o lectura de información respecto a caracteristicas como: consumos, ventas, validez y recargas. En estos bloques de acuerdo al valor de la variable global state que define el boton presionado, se asigna una funcion que genera cambios en las variables antes definidas o cambio de bandera, pero esto bajo la verificación del formato de tres strings que se tiene cuando la manilla es valida como se menciono en la seccion de criterios importantes, generando ademas en el txtOutput un texto que confirma o muestra la actividad que se esta realizando.

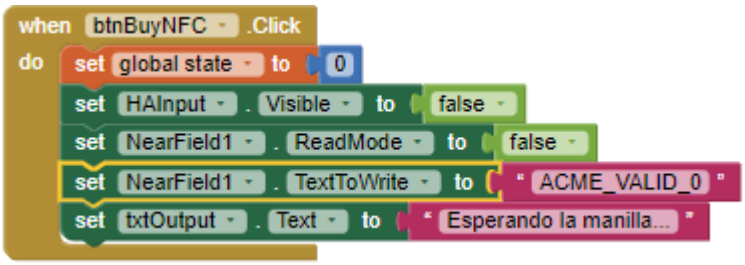
Escritura:

Lectura:

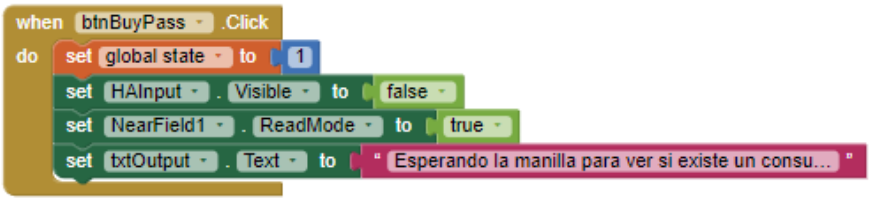
b. Funciones:

Es importante aclarar que estas funciones están relacionadas con los bloques NearField1 mostrados anteriormente.

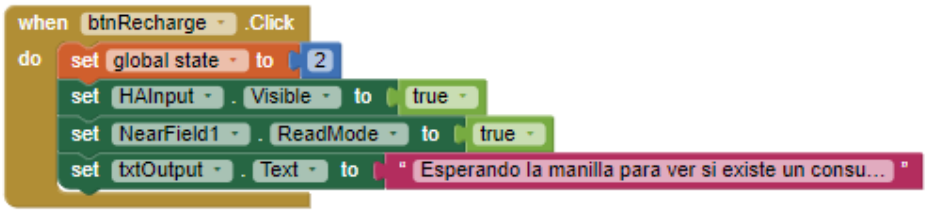
Botón 1 - Comprar manilla (func 1): espera a encontrar la manilla, escribe ACME_VALID_0. Muestra un mensaje en pantalla cuando ha escrito. Incrementa un global 'manillas_vendidas' (bloque NFC).



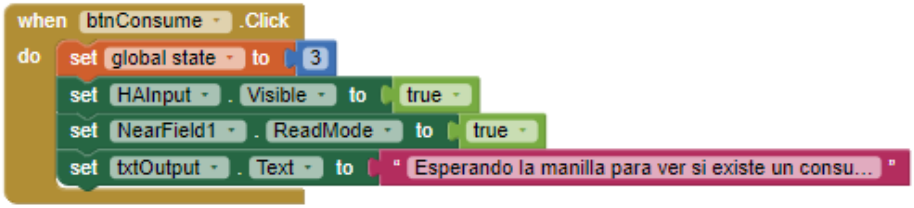
Botón 2 – Comprar entrada (func 2): tan solo espera a escribir ACME_ENDGAME_X. Muestra un mensaje en pantalla cuando ha escrito. Verifica si la persona tiene algún consumo sobrante de un evento anterior y lo reescribe (en la 'X'). Verifica que sea una manilla válida ACME. Incrementa un global 'pases_endgame_vendidos' (bloques Nerfield1).



Botón 3 – Recargar consumo (func 3) lee el estado o pase del evento y lo guarda. Verifica que sea una manilla válida ACME. Lee el valor de consumo en la manilla y lo guarda. Lee un valor de recarga de una entrada de texto y, al escribir el NFC, toma el estado y el valor de consumo anterior (que adiciona al leído). Muestra un mensaje en pantalla cuando ha escrito. Incrementa un global 'consumo_recargado' (bloques NearField1).

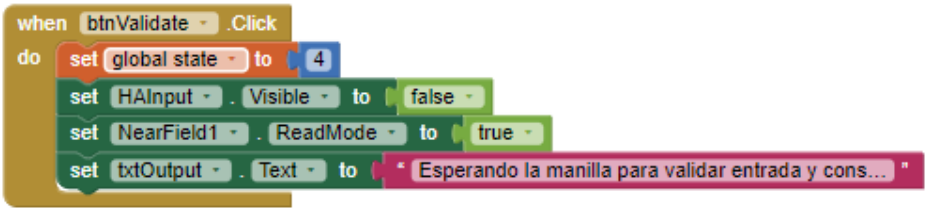


Botón 4- Consumir (func 4): hace lo mismo que el botón 3, pero sustrae el valor ingresado del registrado en la manilla e incrementa un global 'consumo_consumido' con el valor sustraído (bloques NearField1).

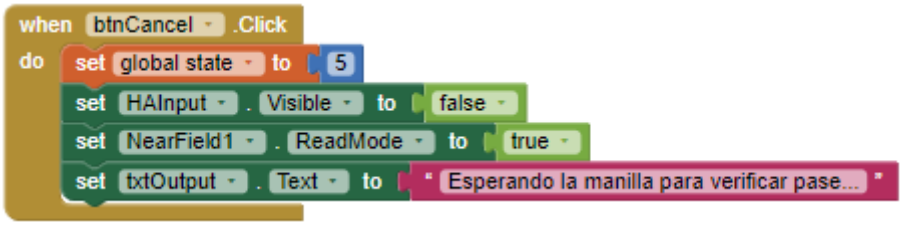


Botón 5 Ver entrada (func 5): lee la manilla y verifica que contenga ACME_ENDGAME o ACME_VALID. Esto indica que tanto la manilla es válida, como que tiene acceso a este evento en específico. Muestra un mensaje que

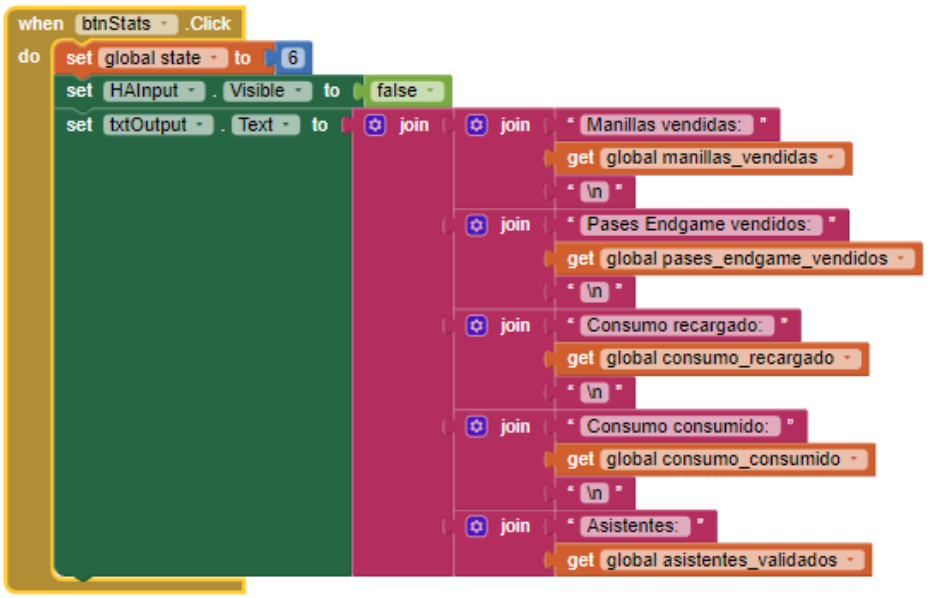
identifica si tiene acceso al evento o si la manilla es válida pero no puede ingresar. Incrementa una global 'asistentes_validados' (bloques NearField).



El botón 6- Cancelar asistencia (func 6) lee la manilla y verifica que contenga ACME_ENDGAME_X. Lee y guarda el valor de consumo X. Escribe en la manilla ACME_VALID_X. Muestra un mensaje en pantalla cuando ha escrito.



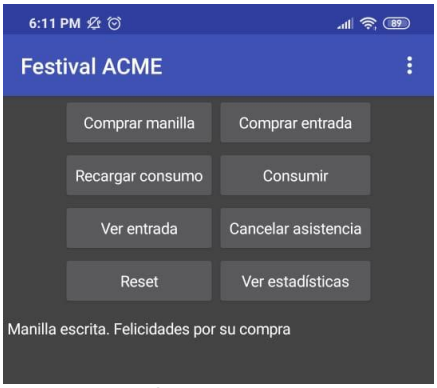
El botón 7- Ver estadísticas (func 7) imprime todas las variables globales.



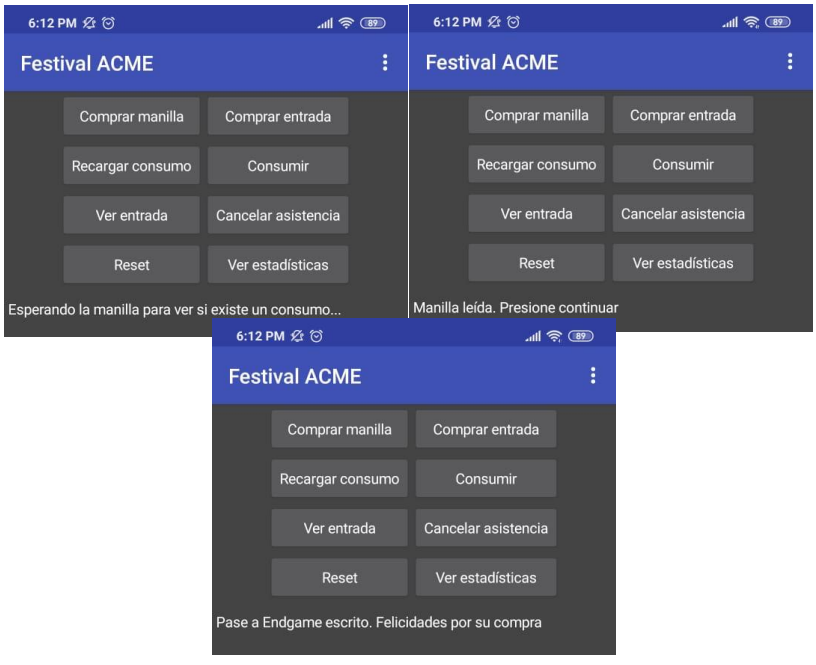
Se adiciono un botón para facilitar las pruebas al cual se le llamo Reset en donde se reinician todas las variables y el proceso

c. Pruebas:

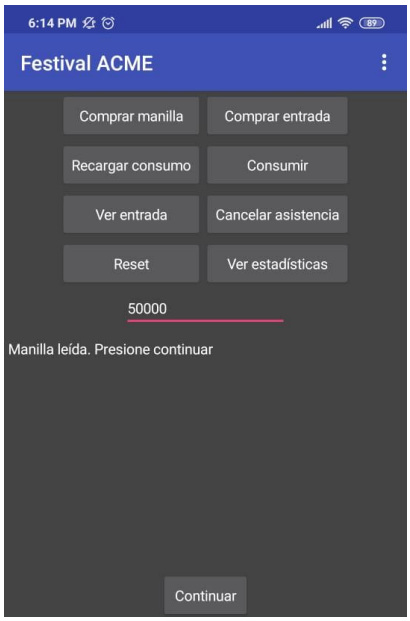
Comprar manilla:



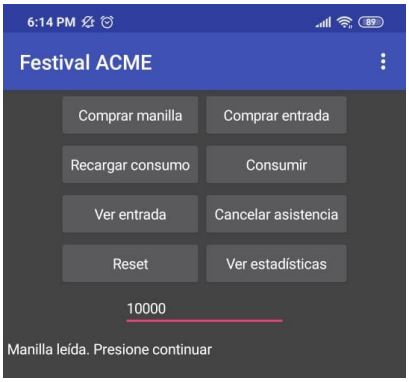
Comprar entrada:

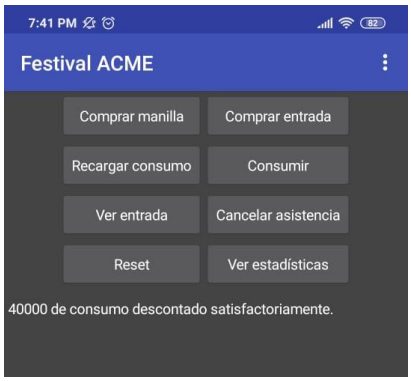


Recargar consumo:



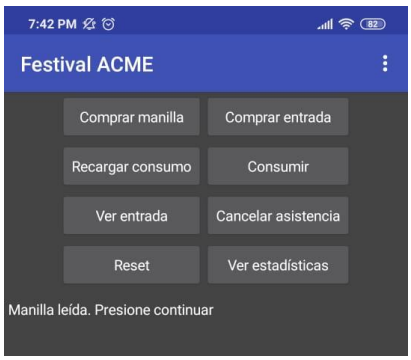
Consumir:

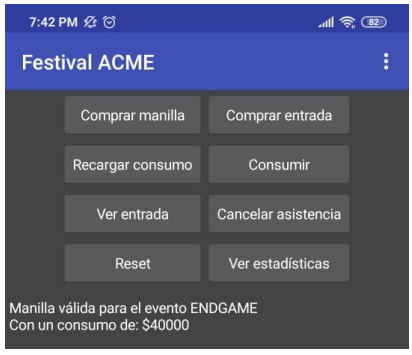




Ver entrada:

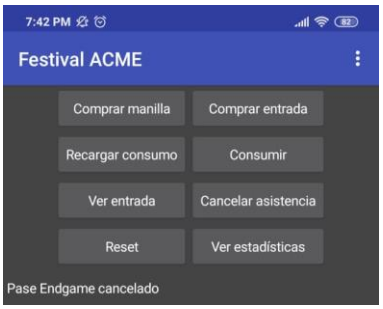
Valida



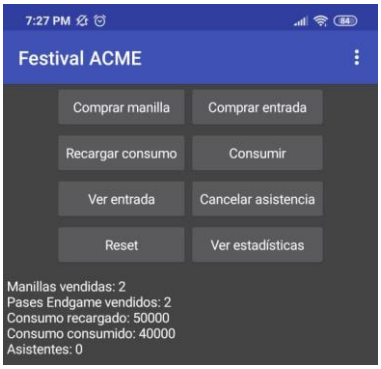


Invalida:

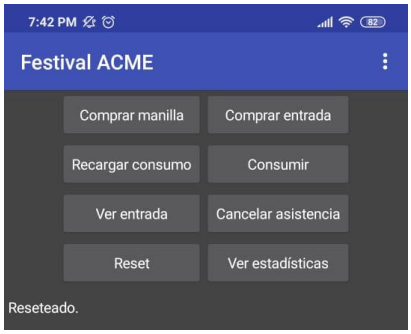
Cancelar asistencia:



Ver estadísticas:

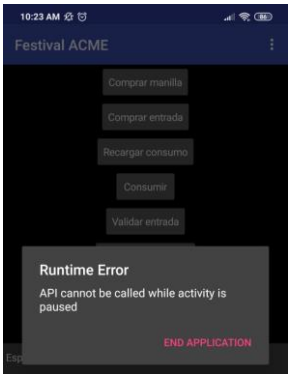


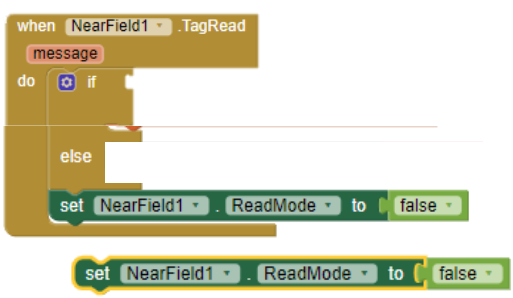
Reset:



d. Errores:

Durante el desarrollo se presentó un error debido a que se estaba cambiando el modo del NFC en pleno evento resultante de la lectura (así que la actividad relacionada al evento no había acabado como para cambiar a escritor NFC). En otras palabras en términos de componentes cuando se finaliza la lectura en nfc.tagRead no es conveniente cambiar el nearField1.ReadMode a False, ya que la aplicación está en medio de una actividad que esta pausada y está mostrando información, así que este cambio generará un conflicto. En la siguiente imagen se ve el error y el bloque que lo produjo el cual hay que eliminar.





No colocar