

TECNOLÓGICO NACIONAL DE MEXICO

Campus La Laguna

Ingeniería en Sistemas Computacionales



GPS

SEMESTRE: Ene – Jun / 2023

PRUEBAS

ALUMNOS:

Carlos Antonio Madrigal Trejo 20130053

Angel Dario Vidaña Vargas 19130984

Omar Adrián Tapia Guzmán 20130022

Carlos Daniel Lopez Romo 19130932

Fernando Perez Romero 19130959

PROFESOR:

Ing. Lamia Hamdan
Torreón, Coah. a 21
de Mayo de 2023

¿Qué es la automatización?

La automatización de aplicaciones es un proceso crucial en el desarrollo de software, que implica la creación y ejecución de pruebas automatizadas para garantizar la calidad y funcionalidad de una aplicación móvil. Al utilizar herramientas y marcos de automatización como Appium, Espresso o UI Automator, los equipos de desarrollo pueden simular interacciones de usuario y escenarios de uso en dispositivos reales o emuladores. Esto permite detectar y corregir rápidamente errores, mejorar la estabilidad y rendimiento de la aplicación, y acelerar los ciclos de desarrollo y lanzamiento. La automatización de aplicaciones también brinda la capacidad de realizar pruebas repetitivas, pruebas de regresión y pruebas de compatibilidad en diferentes dispositivos y versiones de sistema operativo. Al optimizar la eficiencia y confiabilidad de las pruebas, la automatización de aplicaciones ayuda a garantizar una experiencia de usuario consistente y de alta calidad en el amplio y cambiante ecosistema de dispositivos móviles.

Espresso

Espresso es un marco de prueba de interfaz de usuario para aplicaciones Android que facilita la automatización de pruebas funcionales. Diseñado para ser simple y eficiente, Espresso proporciona una API intuitiva y concisa que permite interactuar con los elementos de la interfaz de usuario de manera fluida. Con Espresso, los desarrolladores pueden escribir pruebas claras y legibles que simulan acciones de usuario, como hacer clic en botones, escribir texto, desplazarse por listas y verificar resultados. Además, Espresso ofrece funciones poderosas como la sincronización automática, lo que garantiza que las pruebas se ejecuten en el momento adecuado y esperen que los elementos de la interfaz de usuario estén en un estado válido. Al automatizar las pruebas con Espresso, los equipos de desarrollo pueden detectar rápidamente problemas en la interfaz de usuario, mejorar la calidad de la aplicación y reducir el tiempo necesario para realizar pruebas manuales repetitivas. En general, Espresso es una herramienta valiosa para el desarrollo de aplicaciones Android, que permite realizar pruebas de interfaz de usuario efectivas y confiables.

Para nuestra aplicación se hizo uso de dicha herramienta para automatizar la inserción de datos al crear una nueva cuenta en nuestra aplicación, para esto se utilizó el siguiente código:

```
package mx.GPS.healthec;

import androidx.test.espresso.Espresso;
import androidx.test.espresso.action.ViewActions;
import androidx.test.espresso.matcher.ViewMatchers;
import androidx.test.ext.junit.rules.ActivityScenarioRule;
import androidx.test.filters.LargeTest;
import androidx.test.runner.AndroidJUnit4;

import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

@RunWith(AndroidJUnit4.class)
@LargeTest
public class RegistroActivityTest {

    @Rule
    public ActivityScenarioRule<RegistroActivity> activityScenarioRule =
        new ActivityScenarioRule<>(RegistroActivity.class);

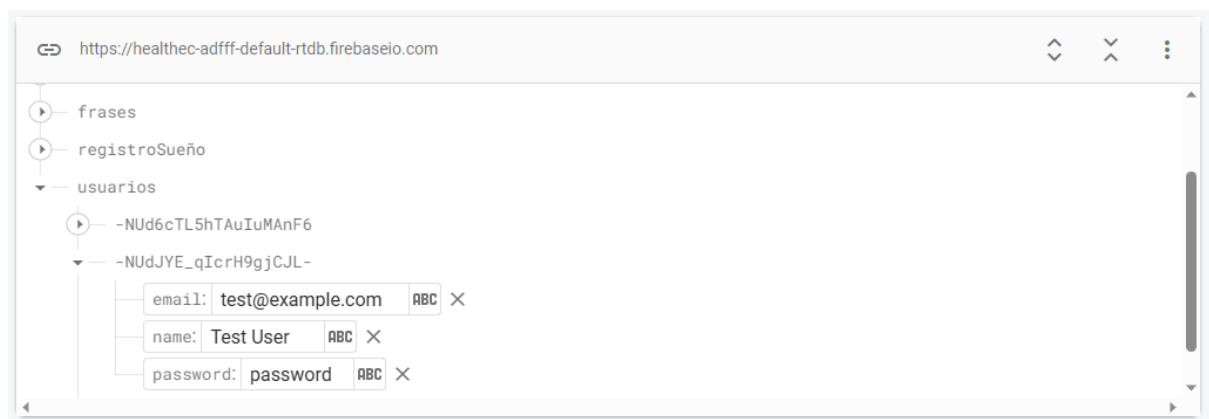
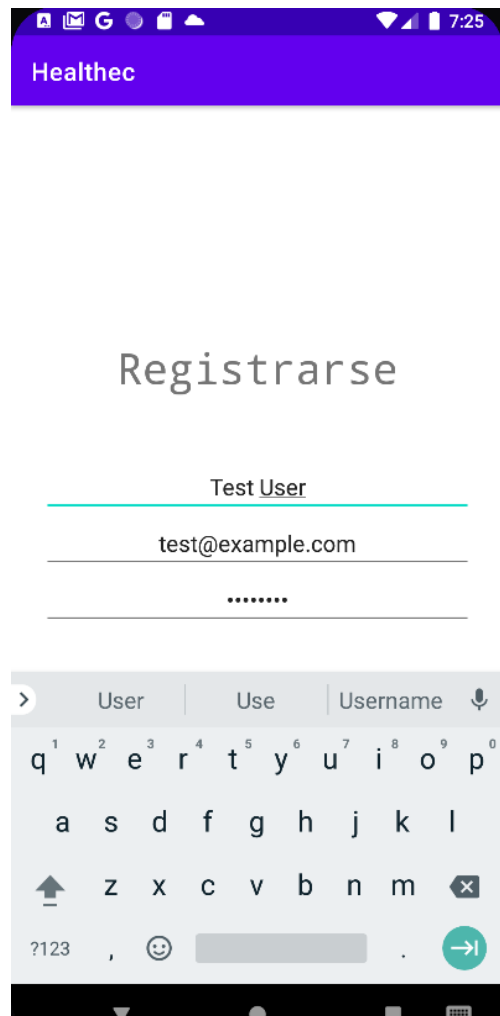
    @Test
    public void testRegistroActivity() {
        // Ingresa el texto en los campos de texto
        Espresso.onView(ViewMatchers.withId(R.id.edt_correoRegistro))
            .perform(ViewActions.typeText("test@example.com"));

        Espresso.onView(ViewMatchers.withId(R.id.edt_passwordRegistro))
            .perform(ViewActions.typeText("password"));

        Espresso.onView(ViewMatchers.withId(R.id.edt_nombreRegistro))
            .perform(ViewActions.typeText("Test User"));

        // Realiza clic en el botón de registro
        Espresso.onView(ViewMatchers.withId(R.id.btn_registroAccept))
            .perform(ViewActions.click());
    }
}
```

Obteniendo los siguientes resultados en la app y base de datos:



Como consecuencia del testeo se pudo realizar una inserción a la base de datos, cumpliendo con la funcionalidad del código lo cual dió como positivo a la prueba de carga de datos, así como esta implementación se puede realizar a otros módulos de la app.