# Estrategia de pruebas

Parcial Final

Pruebas Automáticas

Por: Camilo Niño Jiménez. Correo: [jc.nino11@uniandes.edu.co](mailto:jc.nino11@uniandes.edu.co)

## Características de la app

|  |  |
| --- | --- |
| Nombre | Car Report |
| Descripción | Con esta app tú obtienes una idea de cuánto cuesta tu auto. Simplemente ingresa el costo después de llenar el tanque y obtén buenos reportes. Estos son:  1. Consumo de combustible 2. Precio del combustible 3. Cantidad de millas recorridas 4. Costo promedio del auto por día, mes y año |
| Características | • Simples cálculos como “¿cuánto cuestan 5 litros de combustible?”. • Manejo de otras entradas/gastos como reparaciones o costos recurrentes como impuestos. • Sincronizar tus datos con Dropbox, Google Drive o WebDAV, exportar/importar datos como CSV y hacer simples copias de seguridad a una tarjeta SD. • Recordatorios relacionados al auto como "inspección general de cada año o después de km". |
| Permisos | * Acceso a redes: usado para sincronizar con Dropbox/Google Drive/WebDAV, si está habilitado. * Manejo de cuentas: usado para sincronización. * Escribir en la tarjeta SD: Usado para leer/escribir copias de seguridad y exportaciones CSV a la tarjeta SD. * Correr desde el inicio: usado para mostrar recordatorios después de que el dispositivo se reinicia. |
| Entorno de ejecución | Android. Compatible con todos los dispositivos |
| Desarrollador | Jan Kühle. Auto & Vehicles |
| Publico | Para todos |
| Licencia | Apache 2.0/ GNU |
| Descarga | https://play.google.com/store/apps/details?id=me.kuehle.carreport&hl=en\_US |

## Contexto de pruebas

|  |  |
| --- | --- |
| Presupuesto de pruebas | No aplica |
| Tiempo disponible para pruebas | 18 horas |
| Tipo de pruebas | De regresión / cobertura y densidad |
| Funcionales/No funcionales | Ambas |
| Contexto de uso | N/A |

## Estrategia de pruebas

### Objetivos

* Crear una suite de pruebas sobre la aplicación Car Report, que se utilizara como oráculo de las pruebas de regresión de las aplicaciones mutadas.
* Ejecutar semi-automática y masivamente la suite de pruebas sobre todas las aplicaciones mutadas.
* Recolectar los resultados de la ejecución de la suite de pruebas sobre cada una de las mutaciones para presentar el informe.

### Niveles y tipos

* Pruebas de aceptación al instalar la aplicación, por medio de pruebas exploratorias manuales a través de un emulador Android.
* Pruebas de unidad después de instalar la aplicación, por medio de pruebas JUnit automáticas a través Android Test Orchestator.
* Pruebas de aceptación después de instalar la aplicación, por medio de pruebas BDT automáticas a través de Calabash.
* Pruebas de aceptación después de ejecutar pruebas BDT, por medio de pruebas VRT a través de Resemble JS.
* Pruebas de sistema después de instalar la aplicación, por medio de pruebas Monkey (eventos aleatorios) automáticas a través del comando ADB Monkey.

### Infraestructura de pruebas

|  |  |
| --- | --- |
| Hardware | Equipo portátil. Inter Core i5. 8GB Ram. 1 TB disco duro |
| Software | Windows 10, Calabash, Ejecutor comandos, Android SDK, Android Test Orchestator, Resemble JS |
| Recurso humano | 1 ingeniero de pruebas |

### Calendario y esfuerzo

|  |  |
| --- | --- |
| Planeación estrategia | 2 horas |
| Creación herramienta ejecución masiva | 6 horas |
| Ejecución herramienta | 4 horas |
| Recolección resultados ejecución | 4 horas |
| Documentación de resultados | 2 horas |