|  |  |  |  |
| --- | --- | --- | --- |
| **Alumno(s):** | Yesenia Lemus Orozco | **Fecha:** | 10/05/2018 |

|  |
| --- |
| OBJETIVO |
| El objetivo principal con este proyecto fue utilizar el servicio de Cordova para hacer una aplicación móvil para Android, construida con tecnologías web, como JavaScript, Bootstrap, css, y HTML5, utilizando los plugins de Camara, status de batería, estatus de red, y vibrar, para consultar estados del teléfono, y verificar que funcionan los plugins que se eligieron haciendo pruebas en distintos telefonos. |

|  |
| --- |
| MARCO TEÓRICO |
| *Cordova:* Apache Cordova es un marco de desarrollo móvil de código abierto. Permite utilizar las tecnologías estándar web como HTML5, CSS3 y JavaScript para desarrollo multiplataforma, evitando el lenguaje de desarrollo nativo cada plataformas móviles. Aplicaciones ejecutan dentro de envolturas para cada plataforma y dependen de enlaces estándares API para acceder a de cada dispositivo sensores, datos y estado de la red.  Plugins: Los plugins son aplicaciones o **software que contienen un grupo de funciones o características**  cordova-plugin-network-information: Este complemento proporciona una implementación de una versión anterior de la API de información de red. Proporciona información sobre la conexión celular y wifi del dispositivo, y si el dispositivo tiene una conexión a Internet.  Cordova-plugin-camera: Este complemento define un objeto global navigator.camera, que proporciona una API para tomar imágenes y para elegir imágenes de la biblioteca de imágenes del sistema.  cordova-plugin-network-information: Este complemento proporciona una implementación de una versión anterior de la API de información de red. Proporciona información sobre la conexión celular y wifi del dispositivo, y si el dispositivo tiene una conexión a Internet.  cordova-plugin-battery-status:Este complemento proporciona una implementación de una versión anterior de la API de eventos del estado de la batería. Agrega los siguientes tres eventos al objeto ventana:  cordova-plugin-vibration: Este complemento se alinea con la especificación de vibración W3C http://www.w3.org/TR/vibration/.Este complemento proporciona una forma de hacer vibrar el dispositivo.  Este complemento define objetos globales, incluido navigator.vibrate.  Aunque en el alcance global, no están disponibles hasta después del evento deviceready. |

|  |
| --- |
| DESARROLLO |
| Lo primero que se debe hacer es instalar node.js y su gestor de paquetes, en su última versión desde la página oficial, cordova, con el comando npm install -g cordova, se instala git, con el comando apt-get install git, Apache desde su página oficial, y se agregan las variables de entorno de Java y Apache.  Despues de crea un nuevo proyecto de cordova con el comando, cordova create hello com.example.hello HelloWorld después se de los plugins, cámara con el comando cordova plugin add cordova-plugin-camera, después el de vibrar, con el comando cordova plugin add cordova-plugin-vibration, también el plugin para consultar el estatus de la batería con el comando cordova plugin add cordova-plugin-battery-status  y el plugin sobre la conexión de red, con el comando cordova plugin add cordova-plugin-network-information.  Después se abre el proyecto con cualquier editor, y en la carpeta wwww se encuentra el archivo index.html, se crea un botón por cada plugin que se agregó, con un id y una clase, después en el archivo index.js se agrega el código de cada plugin.    Se agrega un escuchador por cada botón.   Se agrega las funciones que necesita cada plugin que están es la documentación oficial de cordova.     Se agrega color a los botones con bootstrap    Se genera el apk con el comando cordova build    Y después se instala en el teléfono y se prueba |
|  |

|  |
| --- |
| CONCLUSIONES |
| Con este proyecto se llega a la conclusión que se deben tener todas las herramientas que se indican en un inicio para evitar errores en la generación del Apk, también se debe reiniciar la computadora después de instalar todo y de agregar las variables de entorno, es recomendable separar el código por capas para identificar mas fácil los errores y para mayor legibilidad del código, se recomienda también probar la aplicación en el emulador de Lanix, y en el teléfono para probar las funcionalidades de los plugins, si se puede en distintos teléfonos para verificar que en todos funciona perfectamente. |

|  |
| --- |
| BIBLIOGRAFÍA |
| <https://cordova.apache.org/docs/es/latest/guide/overview/>  <https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-network-information/index.html>  <https://cordova.apache.org/docs/es/latest/guide/cli/>  <https://cordova.apache.org/docs/es/latest/guide/platforms/android/index.html>  <https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-device/index.html> |