



Universidad
Nacional
de Quilmes

Laboratorio de sistemas operativos y redes

Trabajo practico:

- Instalación de software libre -
- React Native

Profesor:

- Mainere Facundo

Alumnos:

- Videla Facundo Khalil
- González Matías Hernan

Indice

Introducción.....	3
Requisitos.....	3
Instalación.....	4
Node JS.....	4
Npm.....	4
Expo.....	5
Watchmen.....	5
Controlar celular con la pc.....	5
ADB.....	5
scrcpy.....	5
Creación de un proyecto React Native.....	6
Generalidades del lenguaje.....	10
Solución de problemas.....	12

Introducción

React Native es un **framework JavaScript** para crear aplicaciones reales nativas para iOS y Android, basado en la librería de JavaScript React para la creación de componentes visuales, cambiando el propósito de los mismos para, en lugar de ser ejecutados en navegador, correr directamente sobre las plataformas móviles nativas, en este caso iOS y Android. Es decir, en lugar de desarrollar una aplicación web híbrida o en HTML5, lo que obtienes al final como resultado es una aplicación real nativa, indistinguible de la que podrías desarrollar con tu código en Objective-C o Java.

React Native usa el mismo paradigma fundamental de construcción de bloques de UI (componentes visuales con los que interacciona el usuario) que las aplicaciones nativas reales de Android e iOS, pero gestiona la interacción entre los mismos utilizando las capacidades de JavaScript y React.

Características

Con esta idea de construcción de aplicaciones React Native nos proporciona las siguientes funcionalidades:

- **Compatibilidad Cross-Platform:** las aplicaciones pueden ser ejecutados tanto en iOS como Android simultáneamente con el mismo código base.
- **Funcionalidad nativa:** las aplicaciones creadas mediante React Native funcionan de la misma manera que una aplicación nativa real creada para cada uno de los sistemas usando su lenguaje nativo propio
- **Actualizaciones instantáneas (para desarrollo y/o test):** con la extensión de JavaScript, los desarrolladores tienen la flexibilidad de subir los cambios contenidos en la actualización directamente al dispositivo del usuario sin tener que pasar por las tiendas de aplicaciones propias de cada sistema y sus tediosos ciclos de procesos obligatorios previos.
- **Sencilla curva de aprendizaje**
- **Experiencia positiva para el desarrollador:** si bien la curva de aprendizaje hemos dicho que es sencilla, también el propio lenguaje nos motiva y ayuda a la hora de la evolución según aumentamos nuestro conocimiento y dominio del mismo. Nos ofrece varias características importantes como, por ejemplo, el *Hot reloading* que nos refresca la app en el momento en que guardamos cambios, y nos ofrece una gran ventaja para el desarrollo y testing de nuevas versiones, como hemos comentado arriba. O el uso del *flexbox layout engine* gracias al cual nos permite abstraernos de muchos de los tediosos detalles de la generación de cada uno de los layouts correspondientes a iOS y Android. Así como el uso del debugger de las herramientas de desarrolladores del navegador Google Chrome, facilitando de sobre manera la tarea de depuración de código.

Requisitos

- **Node JS**
 - Node.js es un entorno Javascript del lado del servidor, basado en eventos. Node ejecuta javascript utilizando el motor V8, desarrollado por Google para uso de su navegador Chrome. Aprovechando el motor V8 permite a Node proporciona un entorno de ejecución del lado del servidor que compila y ejecuta javascript a velocidades increíbles. Node es un entorno de código abierto y multiplataforma, por lo que es posible ejecutar Node.js sin ningún tipo de restricción en Windows, Linux y Mac OS X.
- **Npm**
 - NPM es un gestor de paquetes para JavaScript. Sirve para instalar y gestionar versiones de paquetes y librerías js.

NPM usa un fichero especial llamado `package.json` en el que se declaran las librerías y sus versiones. Esto es muy útil ya que puedes tener este archivo con todas las librerías que necesites para que con un simple comando se descarguen todas y no tengas que estar buscándolas en sus respectivos repositorios

- Expo
 - Expo es un framework y plataforma para aplicaciones basadas en React. Es un conjunto de herramientas y servicios creadas entorno a React Native y plataformas nativas que lo ayudan a desarrollar, construir, implementar e iterar rápidamente en iOS, Android y aplicaciones web desde la base de código JavaScript / TypeScript
- Watchmen
 - Watchmen se usa para observar archivos y grabar cuando cambian. También puede desencadenar acciones (como la reconstrucción de archivos) cuando cambian los archivos coincidentes. Básicamente sera utilizado para actualizar los archivos en servidor automáticamente cuando estos tengan cambios.
- Controlar celular con la pc
 - ADB
 - Android Debug Bridge es un software a través del cual se puede darle ordenes especiales al teléfono celular usando comandos de texto. Estos comandos incluyen ordenes básicas que se usan en Linux y una variedad de comandos específicos para desarrolladores.
 - Scrcpy
 - Proporciona visualización y control de dispositivos Android conectados por USB (o sobre protocolo TCP/IP). No requiere ningún acceso a root y funciona en Linux, Windows y masOS. Con este software no queda nada instalado en el dispositivo móvil.

Instalación

En esta sección se expondrán los comandos necesarios para instalar todos los requisitos antes mencionados y algunas particularidades de la instalación

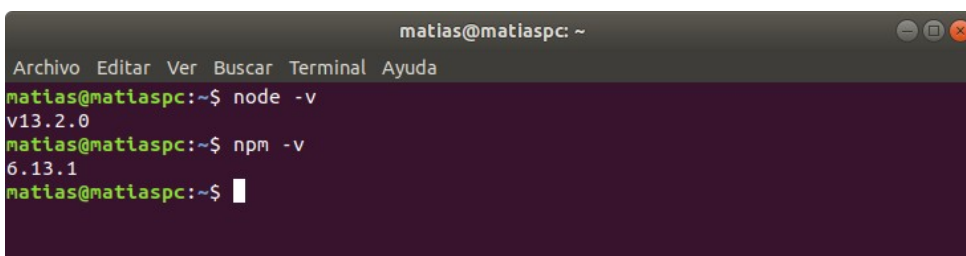
Node JS

- `sudo apt-get install curl`
- `curl -sL https://deb.nodesource.com/setup_13.x | sudo -E bash -`
- `sudo apt-get install -y nodejs`

Npm

- Npm viene junto con la instalación de Node JS

Para verificar que la instalación de Node JS y Npm haya sido correcta se puede verificar la versión de ambos



```
matias@matiaspc: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
matias@matiaspc:~$ node -v  
v13.2.0  
matias@matiaspc:~$ npm -v  
6.13.1  
matias@matiaspc:~$
```

Expo

- `npm install -g expo-cli`

Watchmen

Abrir una consola y posicionarse en la ruta donde tenemos nuestra aplicación. Luego ejecutar los siguientes comandos:

- Descargamos el repositorio
 - `git clone https://github.com/facebook/watchman.git`
- Nos posicionamos en la carpeta watchman y nos pasamos al branch que queremos. En este caso instalamos el branch v4.9.0
 - `git checkout v4.9.0`
- Instalamos y compilamos la aplicación
 - `sudo apt-get install -y autoconf automake build-essential python-dev`
 - `./autogen.sh`
 - `./configure`
 - `make`
 - `sudo make install`

Controlar celular con la pc

ADB

- `sudo add-apt-repository ppa:phablet-team/tools && sudo apt-get update`
- `sudo apt-get install android-tools-adb android-tools-fastboot`

Luego conectar el celular a la pc mediante el cable USB. Para ver los dispositivos conectados se hará mediante el siguiente comando

- `adb devices`

scrcpy

Descargar scrcpy desde su repositorio

- `https://github.com/Genymobile/scrcpy/tree/v1.11`

Compilar la aplicación manualmente y proceder con la instalación

- `sudo apt install ffmpeg libsdl2-2.0-0`

- `sudo apt install gcc git pkg-config meson ninja-build \libavcodec-dev libavformat-dev libavutil-dev \-dev`
- `sudo snap install screpy`

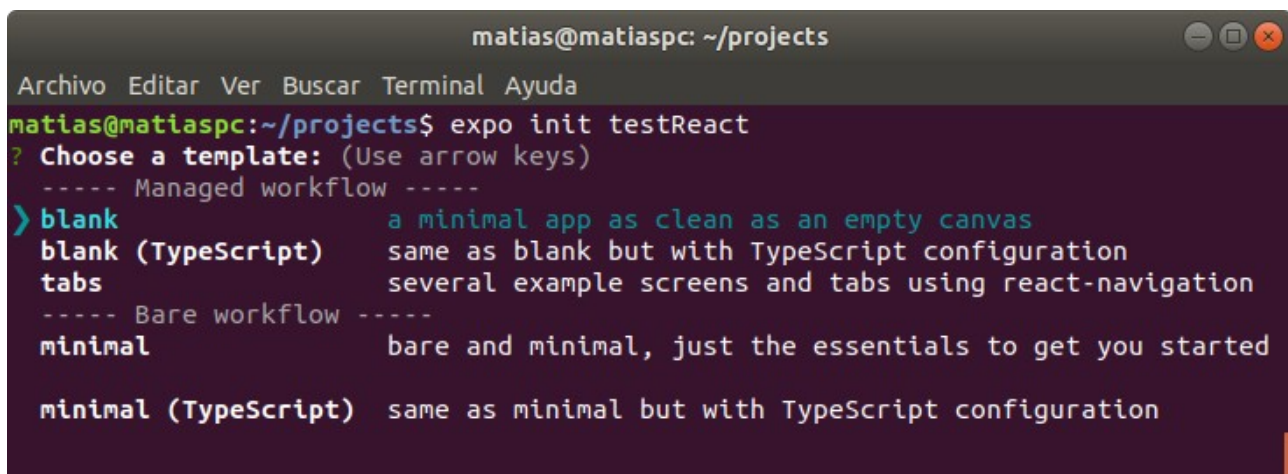
Creación de un proyecto React Native

Expo client ofrece una manera muy sencilla de crear proyectos React Native. Para esto sera necesario ejecutar el siguiente comando que crea el proyecto y toda la estructura de archivos inicial necesaria.

- `expo init projectName`

Para el ejemplo elegimos el nombre “testReact”.

Luego de ejecutar el comando, expo nos ofrece algunos templates para la creación del proyecto:



```
matias@matiaspc: ~/projects
Archivo Editar Ver Buscar Terminal Ayuda
matias@matiaspc:~/projects$ expo init testReact
? Choose a template: (Use arrow keys)
  ----- Managed workflow -----
  > blank          a minimal app as clean as an empty canvas
  blank (TypeScript)  same as blank but with TypeScript configuration
  tabs              several example screens and tabs using react-navigation
  ----- Bare workflow -----
  minimal          bare and minimal, just the essentials to get you started
  minimal (TypeScript)  same as minimal but with TypeScript configuration
```

Templates:

- Managed workflow
 - blank
 - Tiene la configuración mínima de react con un saludo inicial
 - blank (TypeScript)
 - Además de la configuración mínima de react tiene toda la configuración necesaria para utilizar TypeScript
 - tabs
 - Trae incorporado un template para la utilización de tabs
- Bare workflow
 - minimal
 - minimal (TypeScript)

Luego nos pedirá que configuremos los nombres con los que la aplicación se mostrará al momento de la publicación

```
matias@matiaspc: ~/projects
Archivo Editar Ver Buscar Terminal Ayuda
matias@matiaspc:~/projects$ expo init blankTemplate
? Choose a template: expo-template-blank
? Please enter a few initial configuration values.
  Read more: https://docs.expo.io/versions/latest/workflow/configuration/ ▶ 50%
completed
{
  "expo": {
    "name": "<The name of your app visible on the home screen>",
    "slug": "blankTemplate"
  }
}
```

Finalmente se crea el proyecto con la configuración elegida en el template

```
matias@matiaspc: ~/projects/blankTemplate
Archivo Editar Ver Buscar Terminal Ayuda
matias@matiaspc:~/projects/blankTemplate$ ls
App.js      assets      node_modules  package-lock.json
app.json    babel.config.js  package.json
matias@matiaspc:~/projects/blankTemplate$
```

Luego de crear el proyecto y situarnos sobre la raíz del mismo, podemos iniciarlo con el comando

- npm start

Se inicia el servicio de node y se genera un código QR para de esta manera poder verlo y testearlo con el teléfono móvil,

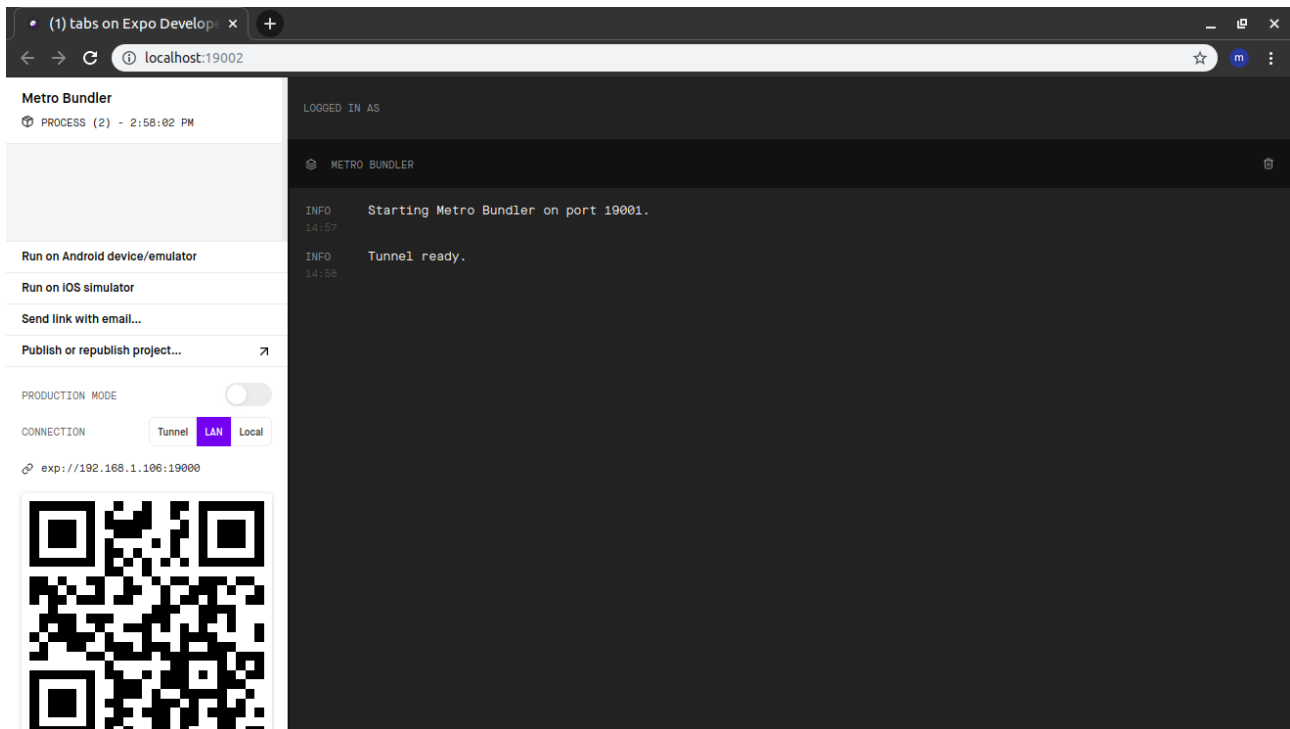


Escaneando ese código QR el celular muestra la aplicación en tiempo real.



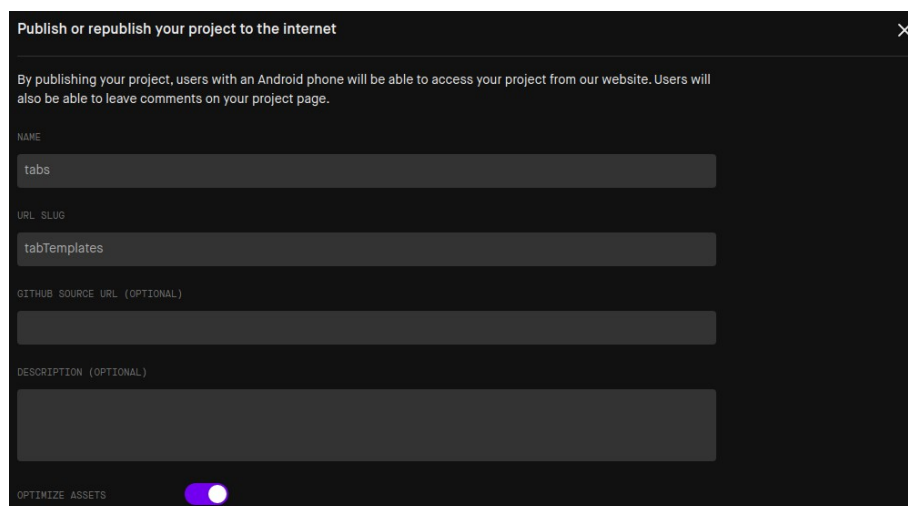
Template tabs

Si elegimos la opción de crear un proyecto de React Native a través del template “tabs” que nos ofrece Expo, luego de iniciar la aplicación con el npm, se nos abrirá un navegador web como muestra la siguiente imagen:



Este es el entorno visual de ejecución de nuestro proyecto en donde se pueden identificar las siguientes secciones:

- Código QR: el mismo que vimos antes, para iniciar la aplicación en un teléfono móvil
- Opciones de emulación: permite iniciar un emulador tanto para android como para iOS
- Compartir: Permite generar el link del proyecto y enviarlo por email
- Tipo de conexión con el teléfono móvil
 - Tunnel
 - LAN
 - Local
- Publicación: permite publicar la aplicación en un repositorio



Publish or republish your project to the internet

By publishing your project, users with an Android phone will be able to access your project from our website. Users will also be able to leave comments on your project page.

NAME
tabs

URL SLUG
tabTemplates

GITHUB SOURCE URL (OPTIONAL)

DESCRIPTION (OPTIONAL)

OPTIMIZE ASSETS

Un menú similar aparece en la consola luego de iniciar la aplicación, solo que sin el entorno gráfico del navegador:

```
› Press a to run on Android device/emulator, or w to run on web.
› Press c to show info on connecting new devices.
› Press d to open DevTools in the default web browser.
› Press shift-d to disable automatically opening DevTools at startup.
› Press e to send an app link with email.
› Press p to toggle production mode. (current mode: development)
› Press r to restart bundler, or shift-r to restart and clear cache.
› Press s to sign out. (Signed in as @ma77ias.)
```

Generalidades del lenguaje

- React-Native no usa HTML para el renderizado de la aplicación, pero proporciona componentes alternativos que funcionan de manera similar. Esos componentes mapean los componentes reales de iOS o Android UI que se representan en la aplicación.
- Los estilos se aplican usando hojas de estilo de una manera similar a lo que se haría en HTML. Por ejemplo:

```
import React, { Component } from 'react';
import { StyleSheet, Text, View } from 'react-native';

const styles = StyleSheet.create({
  bigBlue: {
    color: 'blue',
    fontWeight: 'bold',
    fontSize: 30,
  },
  red: {
    color: 'red',
  },
});

export default class LotsOfStyles extends Component {
  render() {
    return (ca
      <View>
        <Text style={styles.red}>just red</Text>
        <Text style={styles.bigBlue}>just bigBlue</Text>
        <Text style={[styles.bigBlue, styles.red]}>bigBlue, then red</Text>
        <Text style={[styles.red, styles.bigBlue]}>red, then bigBlue</Text>
      </View>
    );
  }
}
```

- React Native utiliza la versión EcmaScript 6 (es6) de javascript
- Utilización de “props”: La mayoría de los componentes se pueden personalizar cuando se crean, con diferentes parámetros. Estos parámetros de creación se denominan props, abreviatura de propiedades.

```
import React, { Component } from 'react';
import { Image } from 'react-native';

export default class Bananas extends Component {
  render() {
    let pic = {
      uri: 'https://upload.wikimedia.org/wikipedia/commons/d/de/Bananavarieties.jpg'
    };
    return (
      <Image source={pic} style={{width: 193, height: 110}}/>
    );
  }
}
```

- Estado: representa a los datos dinámicos de la pagina. Es decir, si tengo algún dato que va a cambiar su valor en la ejecución, debe estar definido en el “state” de la pagina.
- Fetch: React Native nos provee de la API FETCH para hacer peticiones vía rest o hacer peticiones de contenido estático (como se hacia con ajax). Básicamente hace un llamado a la api que se desee y mediante una promesa espera el resultado.

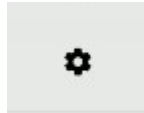
```
function getMoviesFromApiAsync() {
  return fetch('https://facebook.github.io/react-native/movies.json')
    .then((response) => response.json())
    .then((responseJson) => {
      return responseJson.movies;
    })
    .catch((error) => {
      console.error(error);
    });
}
```

Solución de problemas

Teléfono en modo desarrolladores

Al momento de enviar la aplicación al teléfono celular para poder ser probada, es muy probable que de un error, ya que el teléfono no esta preparado para la depuración USB. Para esto simplemente hay que activar la depuración USB en el celular. Seguir los siguientes pasos

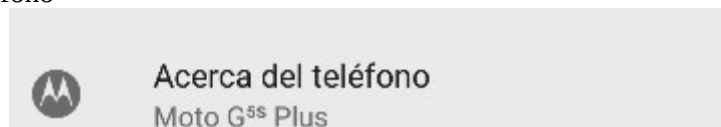
1. Configuración



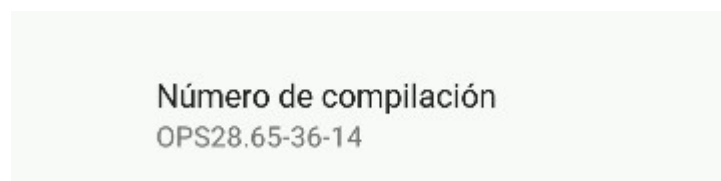
2. Sistema



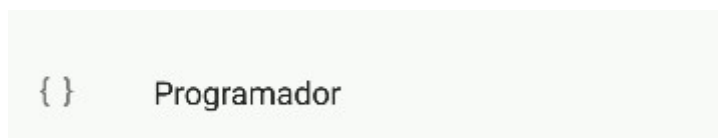
3. Acerca del teléfono



4. Número de compilación (presionar 7 veces)



5. Luego del paso anterior quedara habilitada la opción de “Programador” en el teléfono en la opción de Sistema (punto 2)



6. Finalmente hay que habilitar la “Depuración por USB”

