|  |
| --- |
| http://www.becas.sep.gob.mx/images/logo.png  TECNOLÓGICO NACIONAL DE MÉXICO  Instituto Tecnológico de Chihuahua II |
| **TITANIUM** |
| **APPCELERATOR** |
|  |
|  |
| 01/05/2016  **UNIDAD III**  Lucio Salvador Graciano Venegas  Javier Escudero Gallegos  Denisse Abril Espinoza Ortiz  Nancy Vanessa Balderrama Mares  **APLICACIONES WEB PARA DISPOSITIVOS MÓVILES** |

CONTENIDO

[ÍNDICE DE ILUSTRACIONES 2](#_Toc449473887)

[INTRODUCCIÓN 4](#_Toc449473888)

[¿QUÉ NECESITAMOS? 5](#_Toc449473889)

[INSTALACIÓN DE TITANIUM STUDIO 5](#_Toc449473890)

[APLICACIÓN EN APPCELERATOR 20](#_Toc449473891)

[CÓDIGO DE LA APLICACIÓN 23](#_Toc449473892)

[APP.JS 23](#_Toc449473893)

[WIN1.JS 23](#_Toc449473894)

[WIN2.JS 25](#_Toc449473895)

[DATABASE.JS 26](#_Toc449473896)

[APLICACIÓN CORRIENDO 28](#_Toc449473897)

[CONCLUSIONES 30](#_Toc449473900)

[LINK EN GITHUB 30](#_Toc449473901)

# ÍNDICE DE ILUSTRACIONES

[Ilustración 1 Página de inicio Appcelerator 5](#_Toc447121103)

[Ilustración 2 página de registro 5](#_Toc447121104)

[Ilustración 3 Opciones de desarrolladores. 6](#_Toc447121105)

[Ilustración 4 6](#_Toc447121106)

[Ilustración 5 7](#_Toc447121107)

[Ilustración 6 7](#_Toc447121108)

[Ilustración 7 8](#_Toc447121109)

[Ilustración 8 Vista de la carpeta donde tenemos guardado el node. 8](#_Toc447121110)

[Ilustración 9 wizard de node 9](#_Toc447121111)

[Ilustración 10 Términos y condiciones node. 10](#_Toc447121112)

[Ilustración 11 dirección del programa 10](#_Toc447121113)

[Ilustración 12 características de instalación 11](#_Toc447121114)

[Ilustración 13 Inicialización de la instalación de node. 11](#_Toc447121115)

[Ilustración 14 Instalación de node. 12](#_Toc447121116)

[Ilustración 15 Finalización de la instalación de node. 12](#_Toc447121117)

[Ilustración 16 Vista de la carpeta donde tengamos guardado el archivo de Instalación 13](#_Toc447121118)

[Ilustración 17 Inicialización de la instalación 13](#_Toc447121119)

[Ilustración 18 Instalación del Appcelerator. 14](#_Toc447121120)

[Ilustración 19 Finalización de la insralación 14](#_Toc447121121)

[Ilustración 20 15](#_Toc447121122)

[Ilustración 21 Vista de escritorio Appcelertor 15](#_Toc447121123)

[Ilustración 22 Programa cargando 15](#_Toc447121124)

[Ilustración 23 Selección de dirección de guardado. 16](#_Toc447121125)

[Ilustración 24 Appcelerator cargando 16](#_Toc447121126)

[Ilustración 25 Vista de los requerimientos. 17](#_Toc447121127)

[Ilustración 26 Vista de la Licencia Appcelerator 17](#_Toc447121128)

[Ilustración 27 Vista de la página para descargar Visual Studio 18](#_Toc447121129)

[Ilustración 28 SDK de Android instalándose. 18](#_Toc447121130)

Ilustración 29 WorkSpace……………………………………………………………………………………………………...…20

Ilustración 30 Appcelerator Studio…………………………………………………………………………………………...20

Ilustración 31 New Project……………………………………………………………………………………………………….21

Ilustración 32 Classic……………………………………………………………………………………………………………….21

Ilustración 33 Project Location…………………………………………………………………………………………………22

Ilustración 34 Editor…………………………………………………………………………………………….………………….23

Ilustración 35 App.js………………………………………………………………………………………………………………..23

Ilustración 36 Código de la Aplicación………………………………………………………………………………………24

# INTRODUCCIÓN

Titanium es un framework desarrollado por Appcelerator, el cual permite desarrollar aplicaciones multiplataforma a partir de tecnologías web como Java Script, HTML, CSS, Ruby, Python y PHP, convirtiendo nuestro código fuente en código nativo para cada plataforma ya sea para Móviles (iOS, Android y BlackBerry) o de Escritorio (Linux, Mac OSX y Windows).

A pesar de que podemos instalar Titanium Studio en estos tres sistemas operativos tenemos las siguientes limitaciones:

-Para Windows y Linux solo podemos desarrollar aplicaciones para Google Android. Esto se debe a que el entorno y las herramientas de desarrollo para Apple iOS solo pueden instalarse en Mac OSX.

-Para Mac OSX podemos desarrollar aplicaciones para Android y iOS debido  a que es posible instalar tanto las herramientas de desarrollo de Apple (XCode) como las de Android (Android SDK).

Para desarrollar en Titanium es necesario utilizar Titanium Studio, que es el IDE que nos permitirá crear nuestro código y probarlo mediante el Titanium SDK y el SDK ya sea de Android o de iOS para generar la aplicación Nativa. Titanium Studio está basado en Eclipse, lo cual nos brinda un potente IDE de desarrollo para nuestras aplicaciones.

Las aplicaciones son desarrolladas principalmente en JavaScript, utilizando la API que Titanium provee en común para las plataformas móviles. Titanium pre compila el JS y lo empaqueta junto con el runtime y las clases nativas iPhone o Android.

# ¿QUÉ NECESITAMOS?

**-El SDK de Titanium**  
Es necesario registrarnos en la Página de Appcelerator, una vez que nos hayamos registrado, tendremos acceso al instalador del SDK de Titanium así como el IDE de desarrollo Titanium Studio. Elegiremos la plataforma para en la cual programaremos la aplicación

**-El SDK de Android**  
Una vez instalado el SDK de Android, descargaremos las plataformas en las que probaremos nuestras aplicaciones, para ello ejecutamos el SDK Manager de Android, que se encuentra en la carpeta de Instalación de Android.

**-El SDK de iOS**

Solo en caso de utilizar Mac.

# INSTALACIÓN DE TITANIUM STUDIO

Primero tenemos que entrar a la página de Appcelerator, nos pedirá nuestro correo electrónico para poder registrarnos. [**http://www.appcelerator.com**](http://www.appcelerator.com)

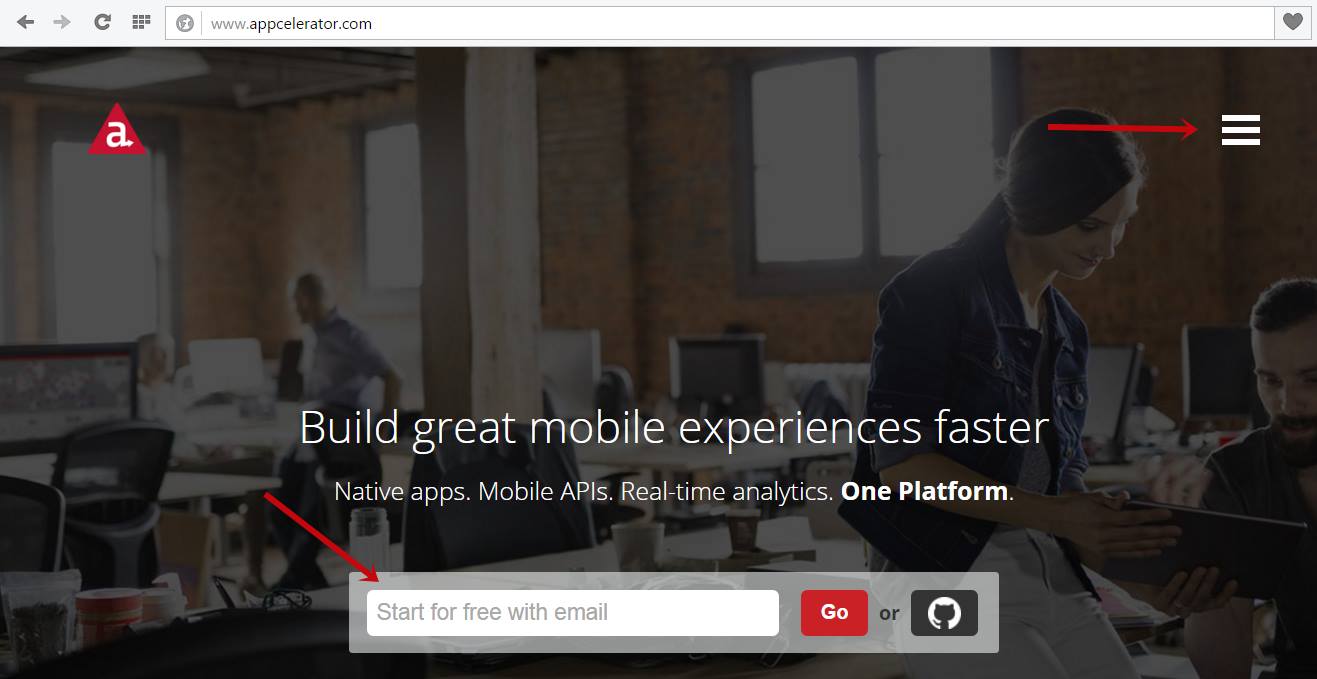


Ilustración 1 Página de inicio Appcelerator

Nos pedirá nuestra cuenta de correo electrónico y una contraseña para crear la cuenta, damos clic en “Sign In”.

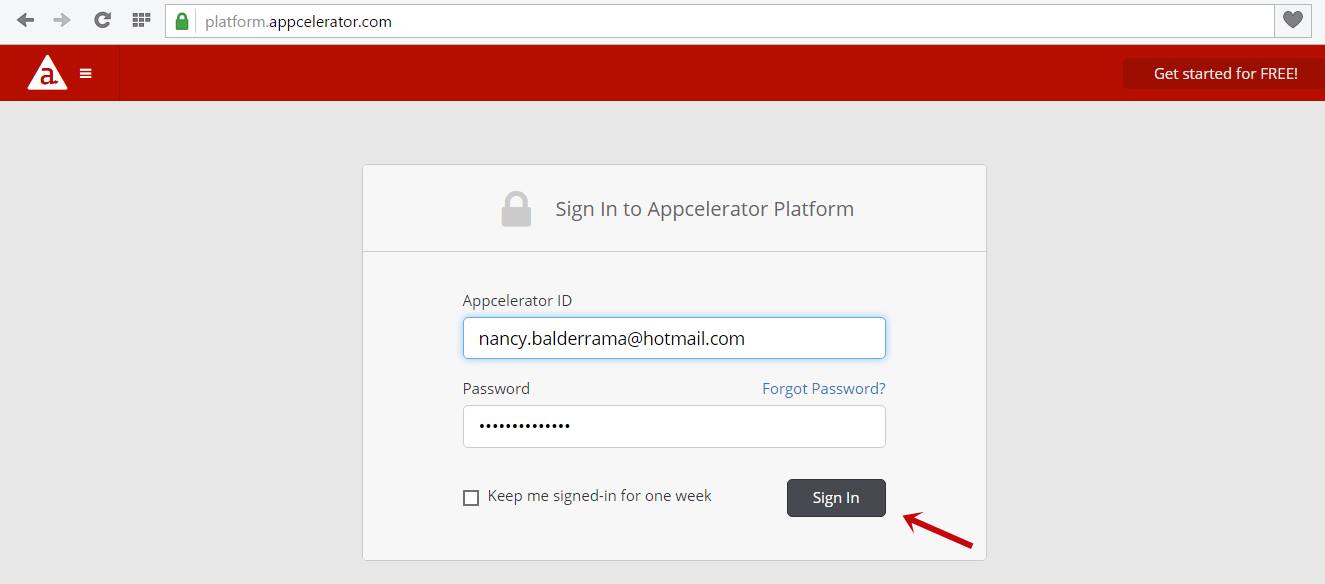


Ilustración 2 página de registro

Hacemos clic en la documentación para desarrolladores. [**http://docs.appcelerator.com**](http://docs.appcelerator.com)

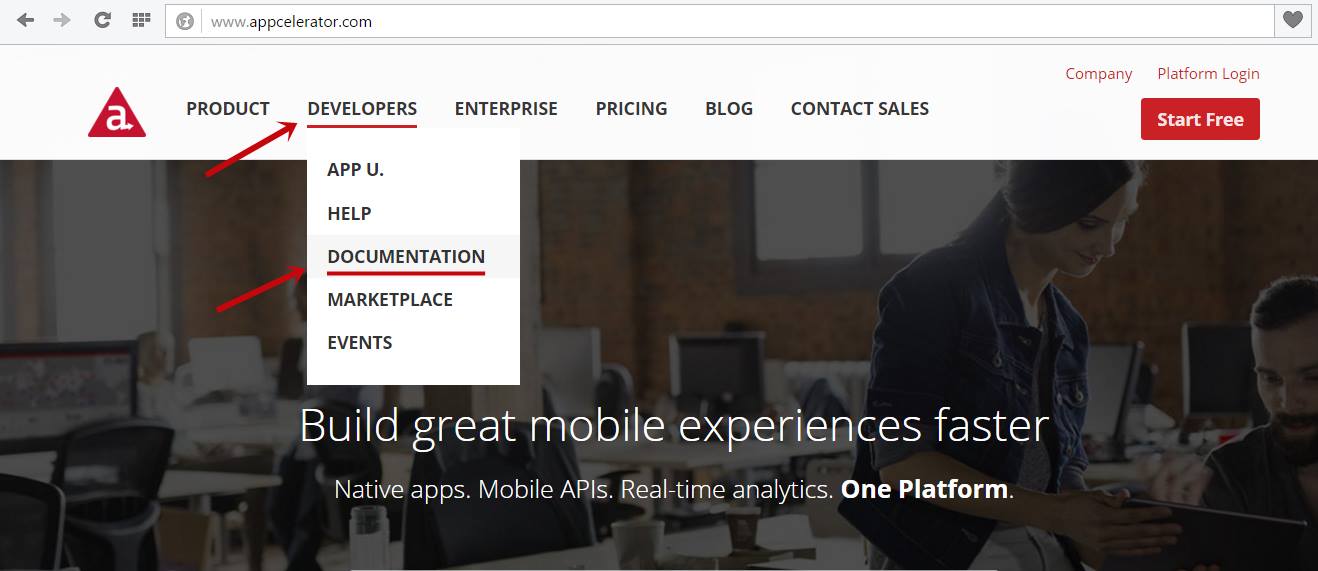


Ilustración 3 Opciones de desarrolladores.

Nos aparecerán cuatro opciónes, damos clic en “Quick Start” (tercera opción de izquierda a derecha, lado superior). Si no funciona de esta manera, también puedes darle clic al logo de Appcelerator ubicado en la esquina superior izquierda y al abrir el menú dar clic en Titanium y se dirigirá a la pantalla de la Ilustración 5.

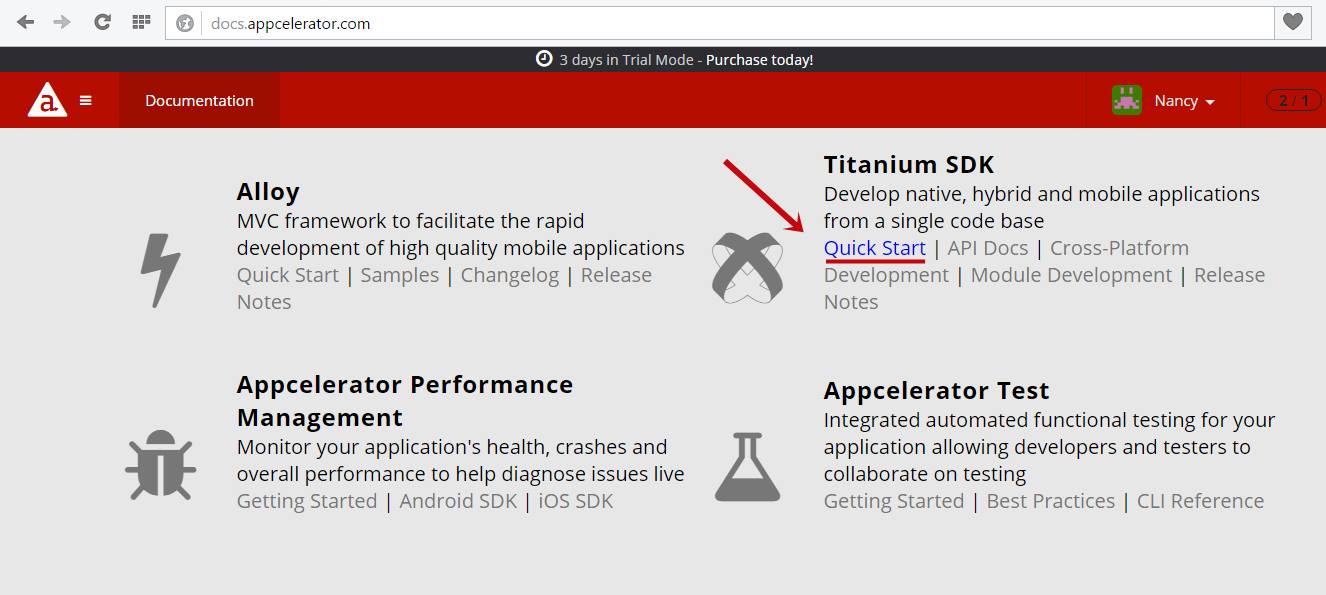


Ilustración 4

Nos mandará a la siguiente página y damos clic en el botón “Get Appcelerator Studio”.

[**https://web.appcelerator.com/product/titanium**](https://web.appcelerator.com/product/titanium)

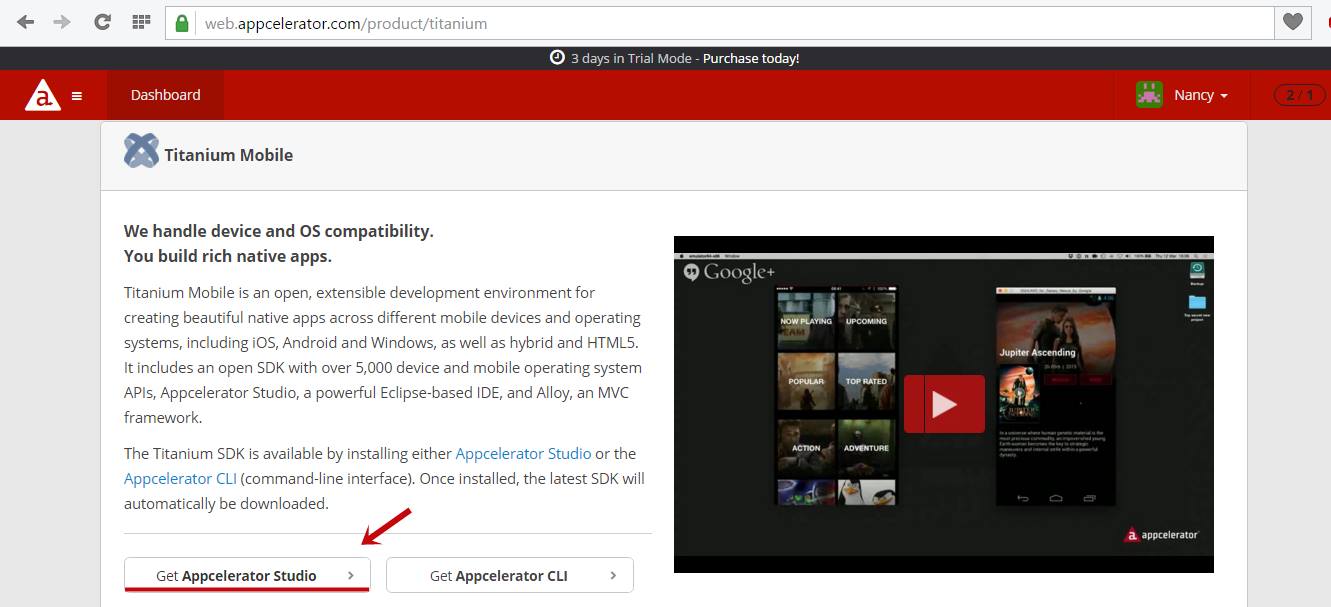


Ilustración 5

En seguida nos aparecerán varias opciones, damos clic en la opción que más nos convenga en este caso para Windows de 32 bits. [**https://web.appcelerator.com/product/studio**](https://web.appcelerator.com/product/studio)

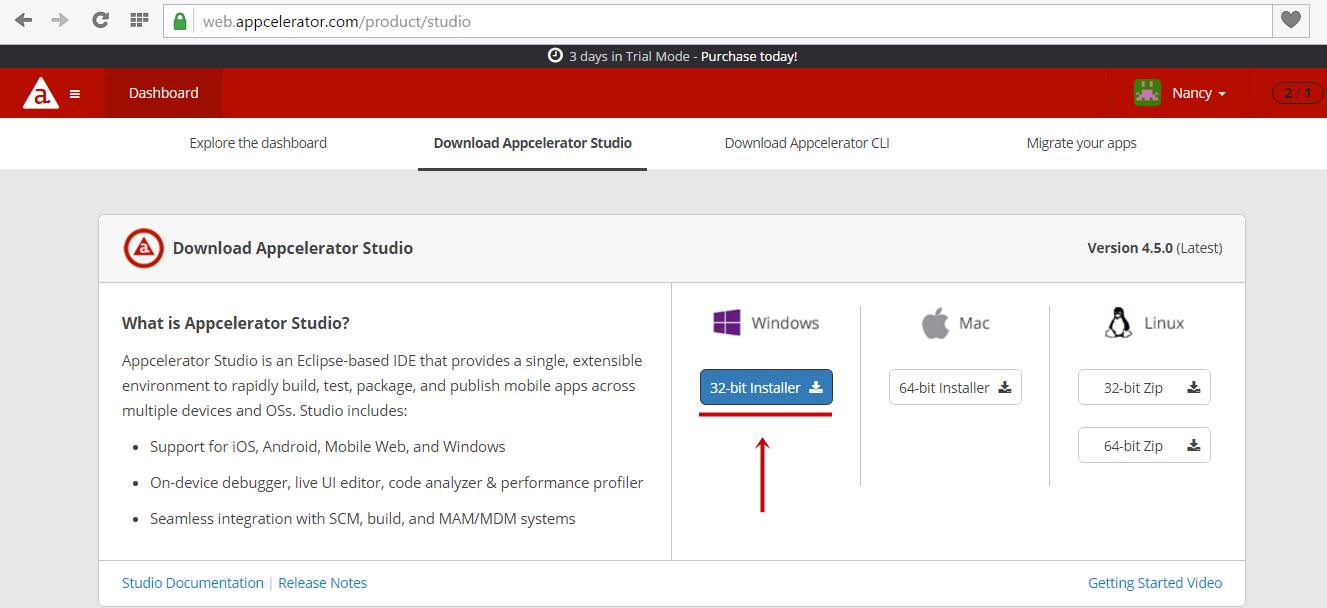


Ilustración 6

Después bajaremos el node para poder trabajar con el Appcelerator Studio.

[**https://web.appcelerator.com/product/cli**](https://web.appcelerator.com/product/cli)

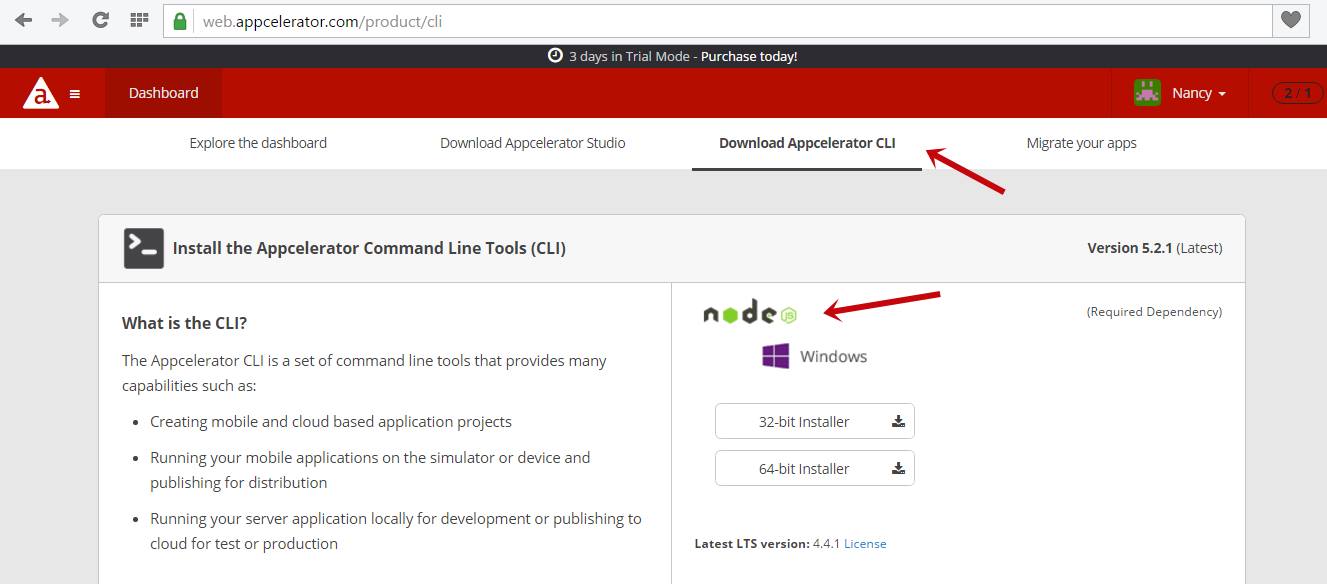


Ilustración 7

Necesitamos instalar primero el intérprete JavaScript node

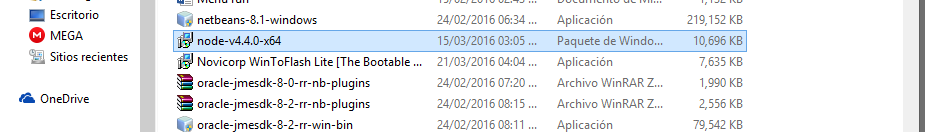


Ilustración 8 Vista de la carpeta donde tenemos guardado el node.

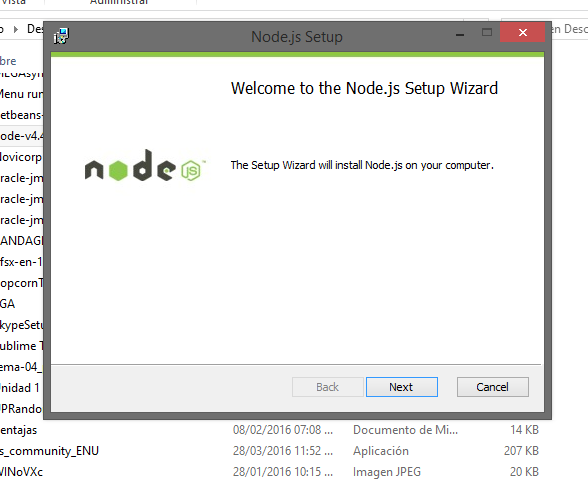
Nos abrirá un Wizard para su instalación, es muy sencillo básicamente es dar clic en siguiente o de acuerdo.

Ilustración 9 wizard de node

Aceptamos los términos y condiciones en la siguiente ventana.

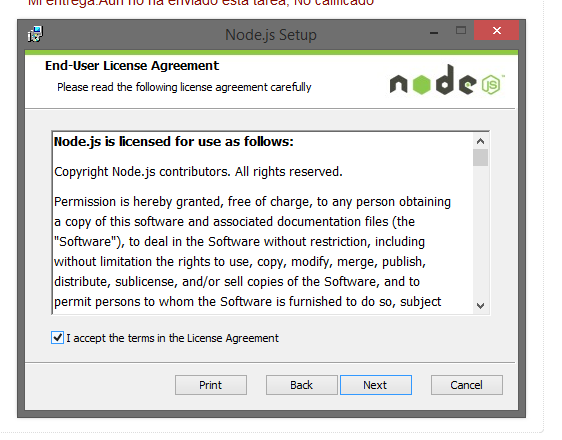


Ilustración 10 Términos y condiciones node.

A continuación nos muestra donde se alojará el programa, damos clic en Next.

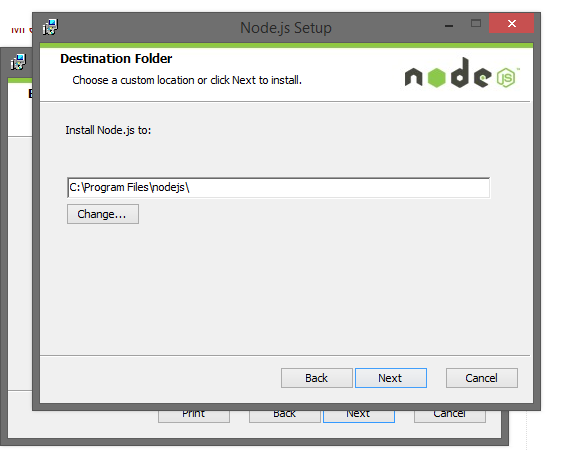


Ilustración 11 dirección del programa

Seleccionamos las características que queramos instalar.

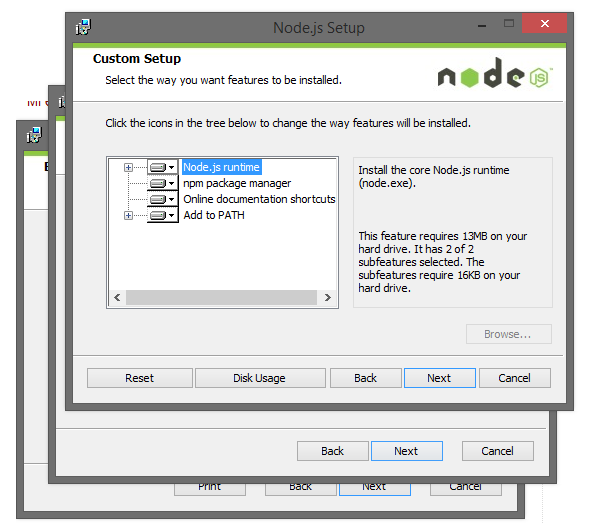


Ilustración 12 características de instalación

Finalmente damos clic en Instalar.

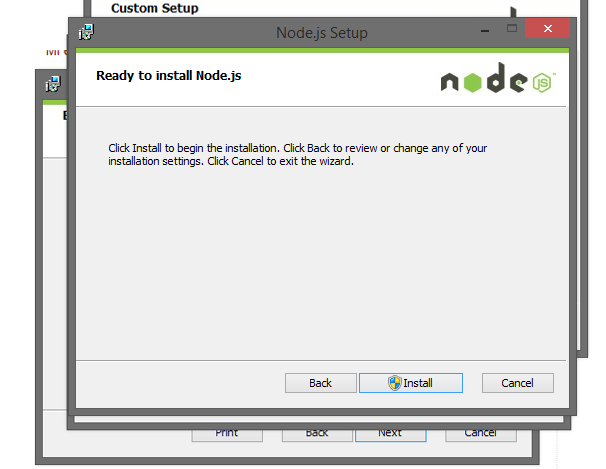


Ilustración 13 Inicialización de la instalación de node.

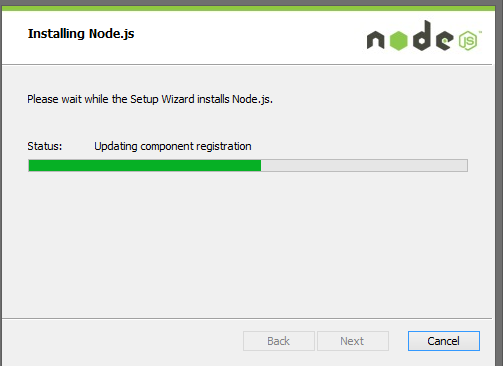


Ilustración 14 Instalación de node.

Clic en “Finish” y enseguida tenemos que instalar el “Appcelerator Studio”

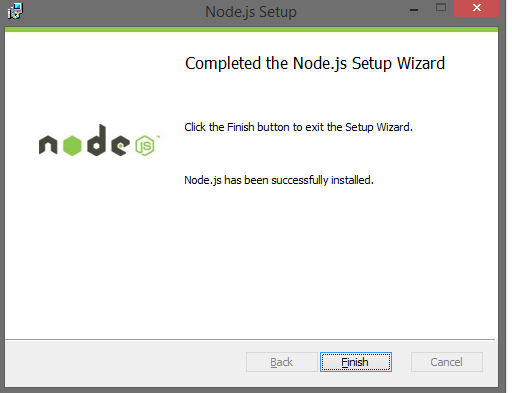


Ilustración 15 Finalización de la instalación de node.

Nos vamos a la carpeta donde guardamos la descarga del Appcelerator.

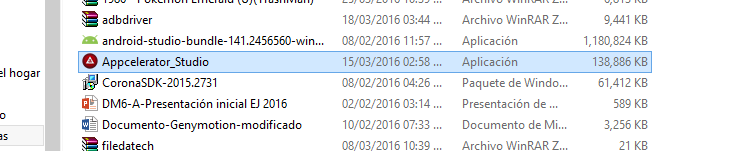


Ilustración 16 Vista de la carpeta donde tengamos guardado el archivo de Instalación

Tenemos que palomear que estamos de acuerdo con la licencia y las condiciones de uso del Appcelerator Studio y damos clic en “Install”

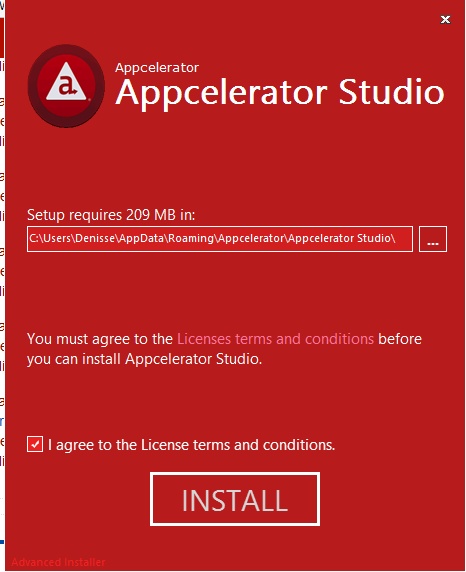


Ilustración 17 Inicialización de la instalación



Ilustración 18 Instalación del Appcelerator.

Damos finish y empezara a cargar el programa.



Ilustración 19 Finalización de la instalación

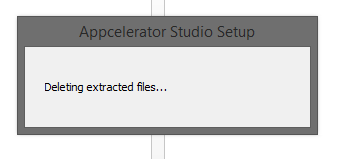


Ilustración 20

Abrimos el Appcelerator Studio para empezar con nuestro primer proyecto.



Ilustración 21 Vista de escritorio Appcelerator



Ilustración 22 Programa cargando

Seleccionamos donde queremos guardar los proyectos que creemos y podemos seleccionar esta dirección por default para futuros proyectos.

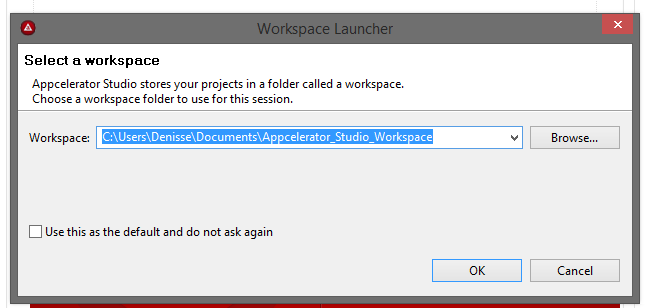


Ilustración 23 Selección de dirección de guardado.



Ilustración 24 Appcelerator cargando

Nos pedirá instalar el SDK de Android y Windows, palomeamos la opción de Android y Windows y clic en configurar.

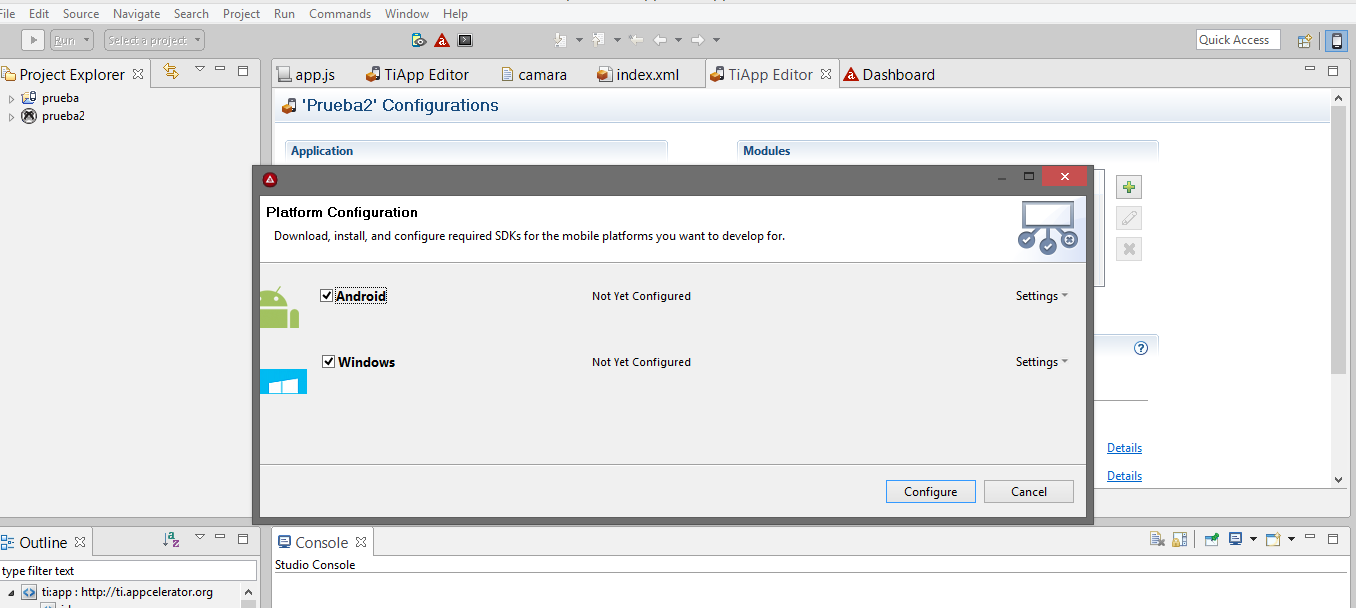


Ilustración 25 Vista de los requerimientos.

Nos abrirá una vista de la las licencias y damos clic en aceptar.

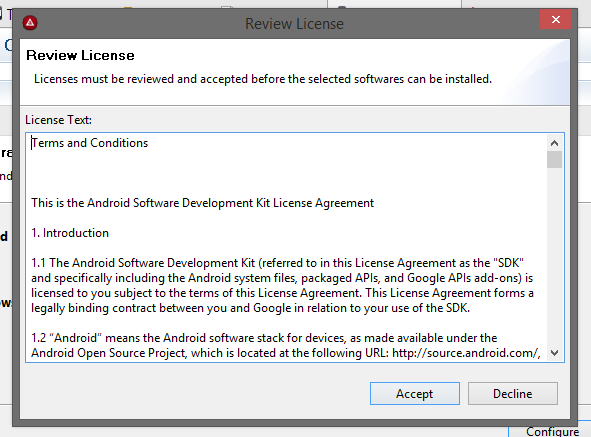


Ilustración 26 Vista de la Licencia Appcelerator

Nos abrirá automáticamente la página de Visual Studio para su descarga, este paso es opcional no es obligatorio descargarlo e instalarlo.

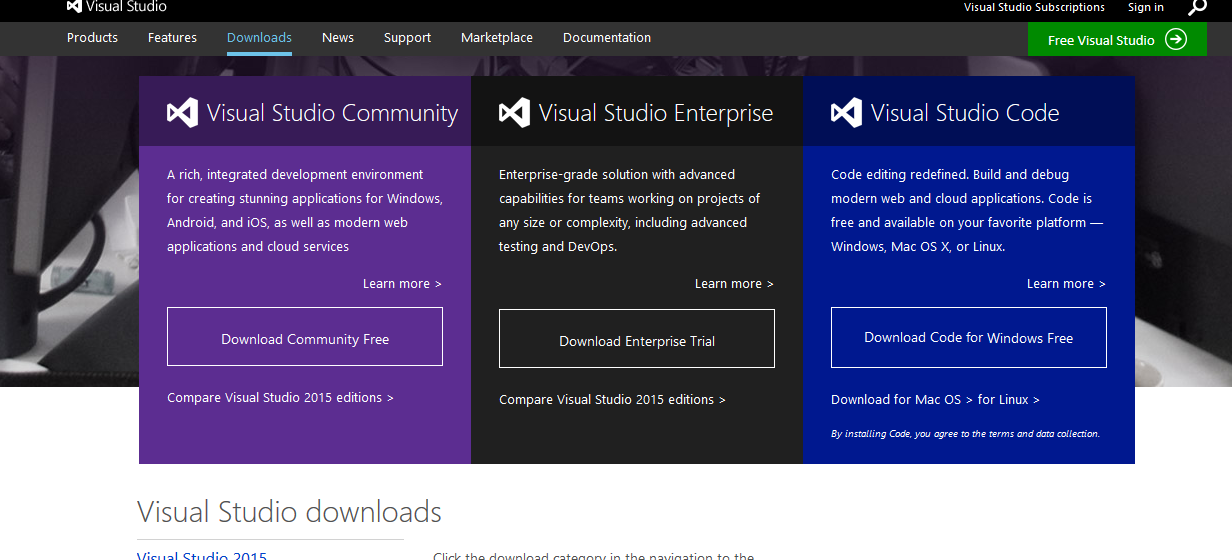


Ilustración 27 Vista de la página para descargar Visual Studio

También empezara la instalación del SDK de Android y cuando finalice comenzará con el de Windows (este proceso es muy tardado y puede durar horas).

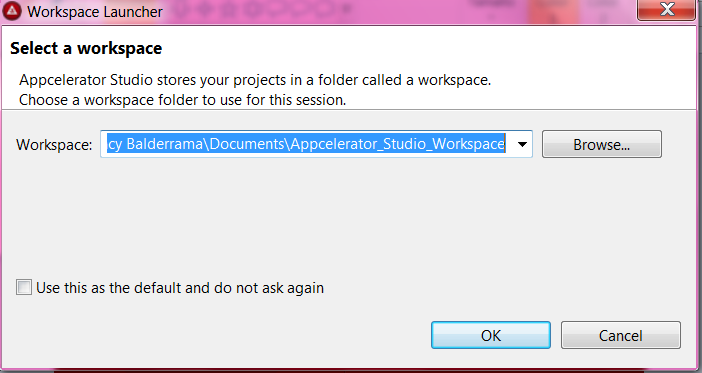


Ilustración 28 SDK de Android instalándose.

Ya cuando termine de descargarse el SDK de Android podremos abrir nuestro Appcelerator y comenzar con nuestra primer práctica.

# APLICACIÓN EN APPCELERATOR

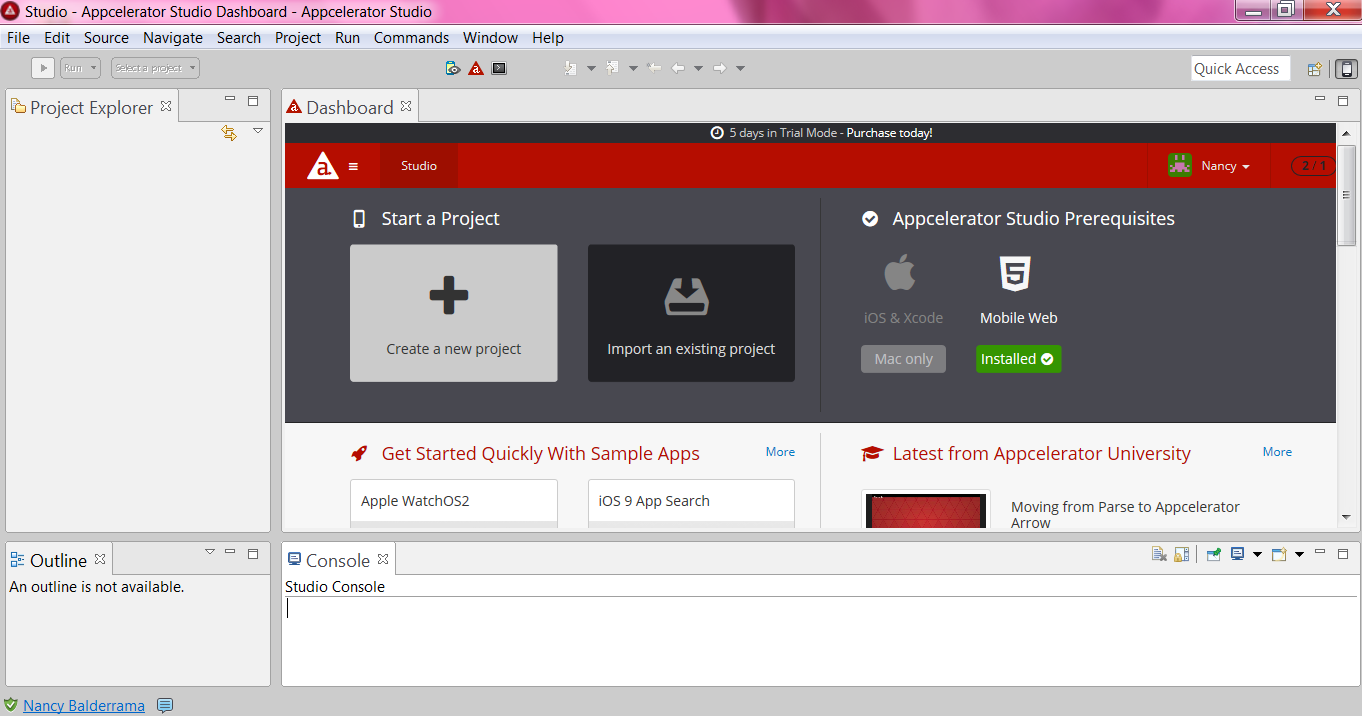
Para comenzar con nuestra primera aplicación abrimos nuestro Appcelerator. Nos aparecerá esta primera opción para guardar nuestras prácticas.



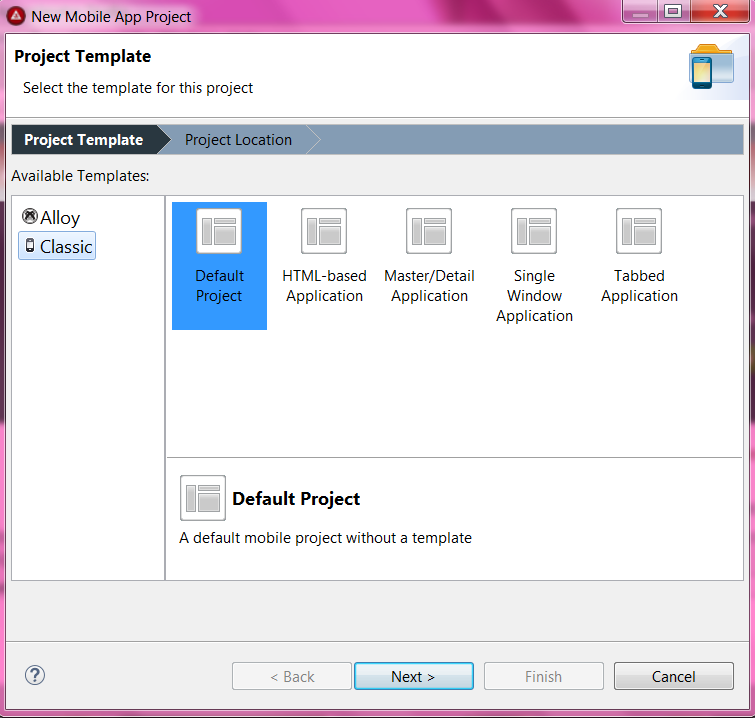
*Ilustración 29 WorkSpace*

Damos clic en OK y empezara a cargarse nuestro Appcelerator.

*Ilustración 30 Appcelerator Studio*

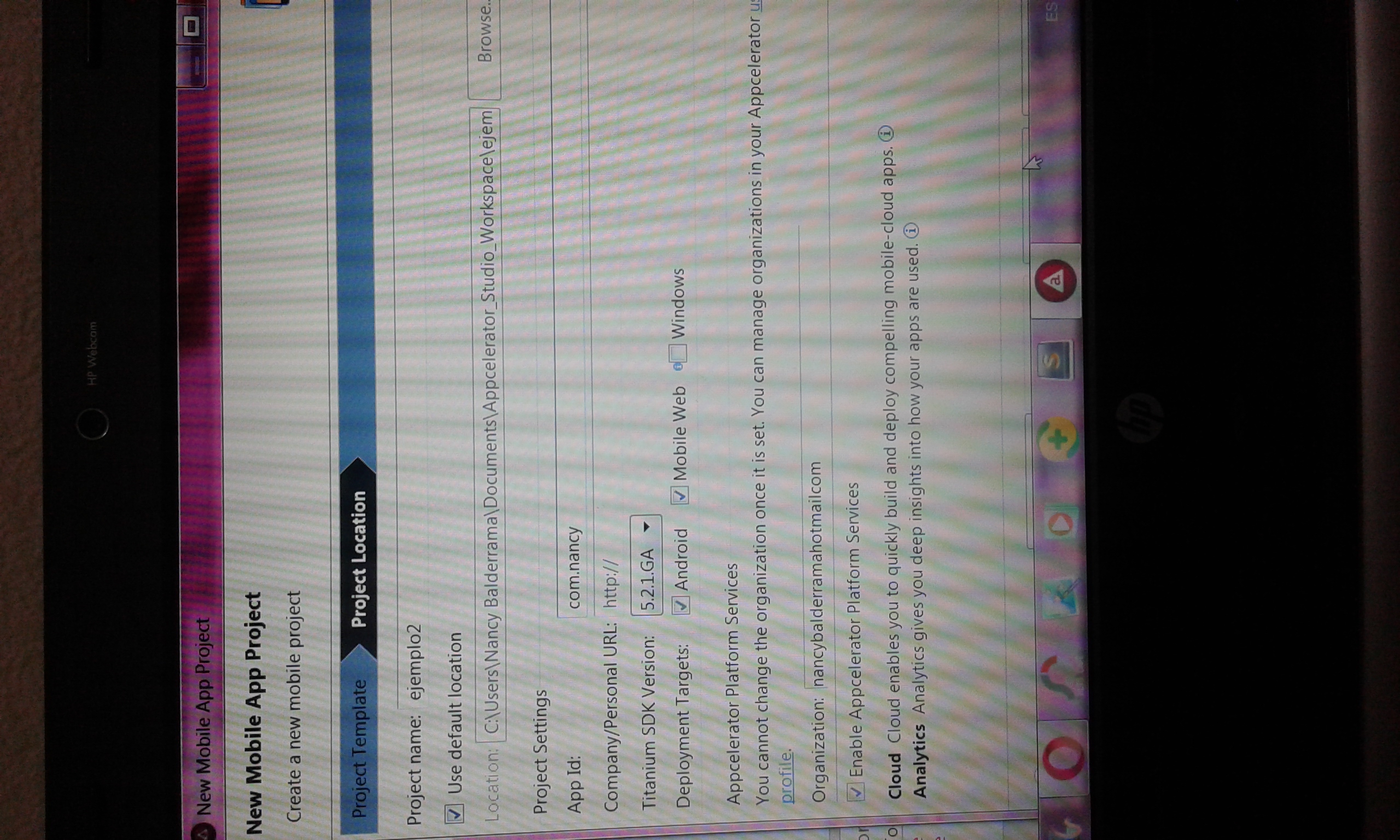
**En la pantalla siguiente selecionaremos créate a new Project.

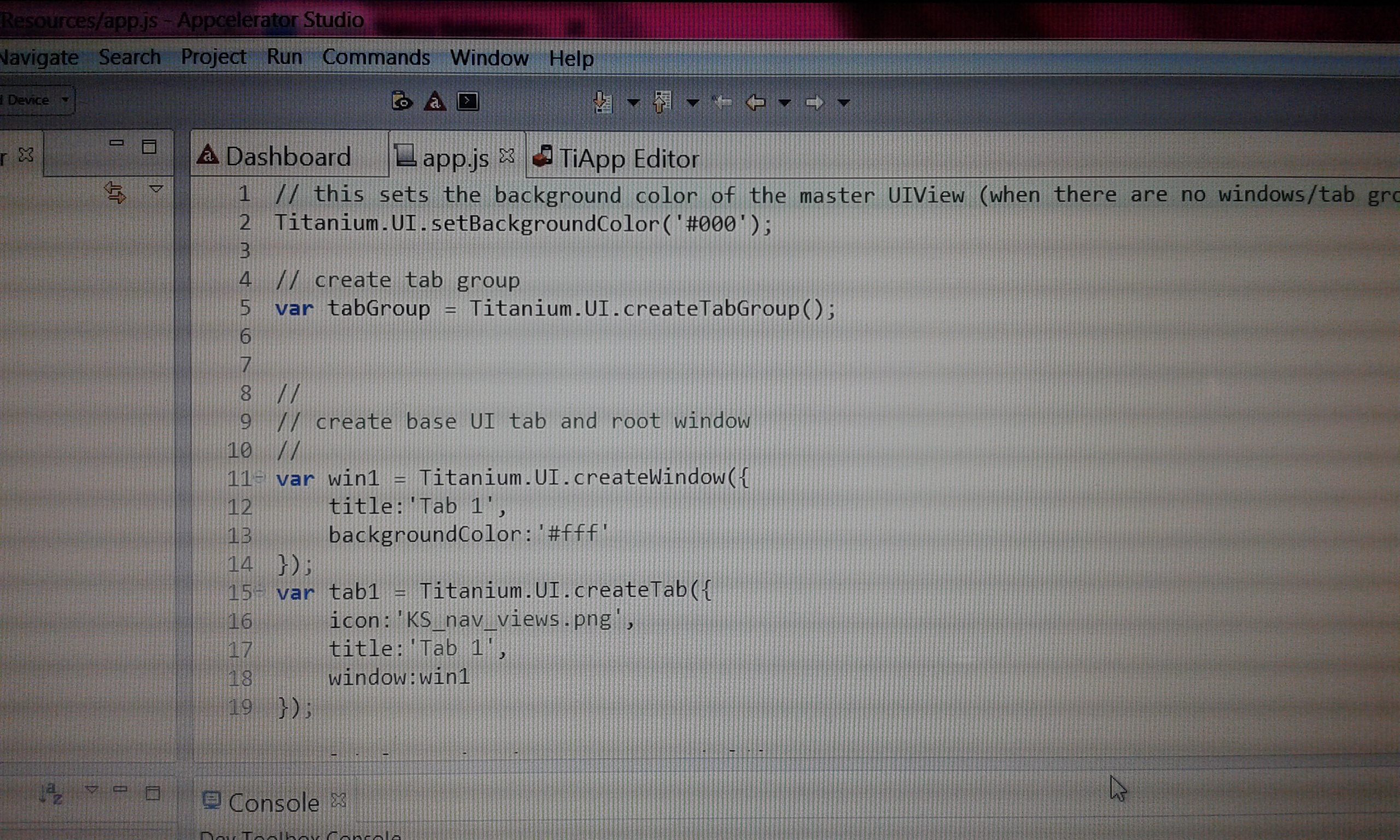
*Ilustración 31 New Project*

Después seleccionamos un proyecto classic y default Project.

*Ilustración 32 Classic*

En la ventana siguiente hay que nombrar nuestra aplicación y en App Id siempre hay que empezar con **com.** Y ya lo que gusten.



Por último se nos abrirá nuestro editor y nos posicionaremos en la pestaña de app.js aquí se nos mostrar código de ejemplo, pero para nuestra aplicación vamos a borrarlo todo para poder empezar a teclear nuestro código.

# CÓDIGO DE LA APLICACIÓN

# APP.JS

//Función Recursiva (SelfCallFunction)

(function(e){

//Creamos un TabGroup

var tabGroup = Ti.UI.createTabGroup(),

//Reservamos memoria para las ventanas y utilizarlas despues

//Se ubican en las carpetas ui

win1 = require('ui/Win1'),

win2 = require('ui/Win2'),

//Creamos 2 Tabs

//Tab1 -> asociamos a la ventana 1

tab1 = Ti.UI.createTab({

title: 'Registros',

window: new win1()

});

//Tab2 -> asociamos a la ventana 2

tab2 = Ti.UI.createTab({

title: 'Tabla',

window: new win2()

});

//Agregamos las pestañas al TabGroup (Contenedor General)

tabGroup.addTab(tab1);

tabGroup.addTab(tab2);

//Abrimos nuestro TabGroup

tabGroup.open();

})();

# WIN1.JS

function Win1(){

//Creamos una ventana

var win = Ti.UI.createWindow({

backgroundColor:'white'

}),

//Creamos una variable para utilizar el archivo DataBase

dataBase = require('ui/DataBase'),

//Creamos un boton con algunas propiedades

//Para crear la base de datos

buttonCreate = Ti.UI.createButton({

title:'create',

width:80,

height:50,

top:0

}),

//Creamos un campo de texto

//Donde ingresamos los datos

textField = Ti.UI.createTextField({

borderStyle: Ti.UI.INPUT\_BORDERSTYLE\_ROUNDED,

color:'black',

width:200,

top:35

}),

//Creamos un boton para insertar los datos en la base de datos

buttonInsert = Ti.UI.createButton({

title:'insert',

width:80,

height:50,

left:50,

top:70

});

//Creamos un boton para borrar los datos de la base de datos

buttonDelete = Ti.UI.createButton({

title:'delete',

width:80,

height:50,

right:50,

top:70

});

//Agregamos todos los elementos a nuestra ventana

win.add(buttonCreate);

win.add(textField);

win.add(buttonInsert);

win.add(buttonDelete);

//Eventos

//Evento para el boton Create

buttonCreate.addEventListener('click', function(e){

dataBase('create',null);

});

//Evento para el boton Insert

buttonInsert.addEventListener('click', function(e){

//Necesitamos agregar el valor del textField

dataBase('insert',textField.value);

});

//Evento para el boton Delete

buttonDelete.addEventListener('click', function(e){

dataBase('delete',textField.value);

});

//Regresamos la ventana, todo lo que agreguemos en el contenedor se podra visualizar

return win;

}

//Para mostrar las ventanas

module.exports = Win1;

# WIN2.JS

function Win2(){

//Creamos una ventana

var win = Ti.UI.createWindow({

backgroundColor:'#000'

}),

dataBase = require('ui/DataBase'),

//Creamos un TableView

table = Ti.UI.createTableView({

});

//Añadimos el TableView a nuestro contenedor

win.add(table);

//Evento

//Cada vez que win tenga el foco se actualiza

win.addEventListener('focus',function(e){

//Creamos una nueva consulta

var dataBaseObj = new dataBase('querying',null);

//Nuestro arreglo de objetos TableViewRow

data = [];

//Mientras i este en dataBaseObj

for (var i in dataBaseObj) {

//Genera una fila

var row = Ti.UI.createTableViewRow({

title: dataBaseObj[i]

});

//Agregamos la fila

data.push(row);

};

//Para mostrar los datos en la Tabla

table.data = data;

});

//Nos regresara la ventana con todos los elementos

return win;

}

module.exports = Win2;

# DATABASE.JS

//Funcion con dos parametros

function DataBase(request,name){

switch(request){

//En caso de que sea el click del boton create

//Aquí solo se usa el parametro request

case 'create':{

//var db = Ti.DataBase.install('ui/DataBase','dataBaseName');

//var db = Ti.DataBase.install('/mydatabase/DataBase','dataBaseName');

//Abre la base de datos

var db = Ti.Database.open('miBaseDeDatos');

//Creamos la tabla con 2 campos (id y name)

db.execute('CREATE TABLE IF NOT EXISTS data ( id INTEGER PRIMARY KEY,name TEXT);');

//Cerramos nuestra base de datos

db.close();

//Mensaje para mostra si se creó la tabla

alert("Tabla creada");

}

break;

//En caso de que sea el click del boton insert

case 'insert':{

var db = Ti.Database.open('miBaseDeDatos');

//Utilizamos el parametro name

db.execute('INSERT INTO data(name) VALUES ("'+name+'")');

db.close();

alert("Dato Insertado");

}

break;

//En caso de que sea el click del boton delete

case 'delete':{

var db = Ti.Database.open('miBaseDeDatos');

//Esta es otra forma de concatenar =?

db.execute('DELETE FROM data WHERE name= ?', name);

db.close();

alert("Elemento Eliminado");

}

break;

//En caso de que se quiera hacer una consulta

case 'querying':{

var db = Ti.Database.open('miBaseDeDatos');

//Guardamos el query

var allData = db.execute('SELECT \* FROM data');

//Guardamos los datos en un Array (Strings)

var data =[];

//Mientras allData tenga Rows validos

while(allData.isValidRow()){

//Obtiene el campo que tiene name

//y lo mete en el arreglo data

data.push(allData.fieldByName('name'));

//Pasa al Row siguiente

allData.next();

}

db.close();

//Regresamos data para utilizarla en la interfaz

return data;

}

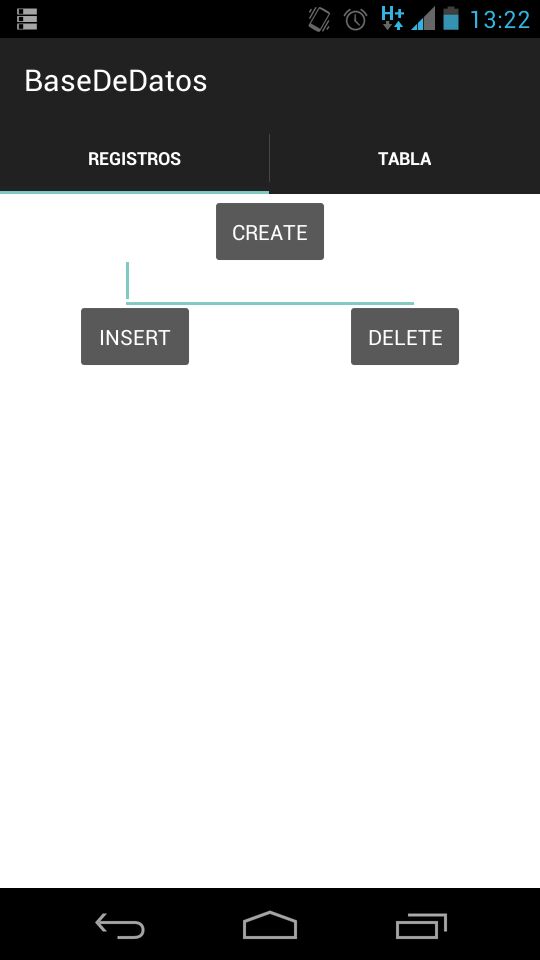
break;

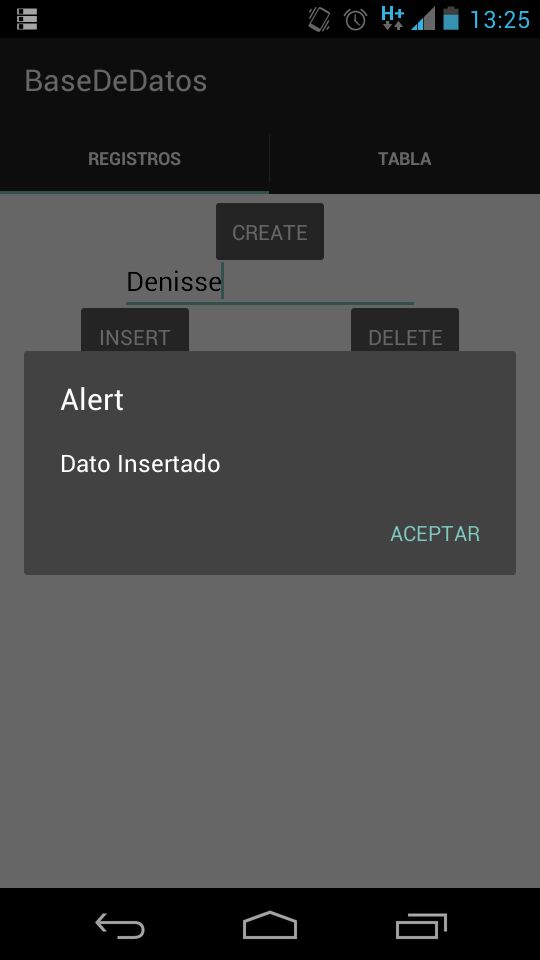
}

}

module.exports = DataBase;

# APLICACIÓN CORRIENDO





# 

# 

# CONCLUSIONES

Al usar este framework nos dimos cuenta que no es tan fácil al momento de la instalación, es algo complejo, pero cuando una vez se logra instalarlo al hacer nuestras aplicaciones no es muy difícil la codificación y no se necesita estar cargando y cargando la aplicación en nuestro cel, ya que si se le modifica algo al código automáticamente se actualiza la aplicación en el celular. Tiene ventajas y desventajas como todos pero si podría ser algo amistoso con el usuario una vez familiarizado con este framework.

# LINK EN GITHUB

**https://github.com/javierescudero/Base-de-datos-Titanium.git**