**UAH**

16

**Desarrollo de aplicaciones multiplataforma**

**TG3**

**Contenido**

[1. Autores del trabajo, planificación y entrega 4](#_Toc450378091)

[1.1. Autores 4](#_Toc450378092)

[1.2. Planificación 4](#_Toc450378093)

[1.3. Entrega 4](#_Toc450378094)

[2. Nombre del prototipo 5](#_Toc450378095)

[2.1. Requisitos funcionales 5](#_Toc450378096)

[2.2. Otros requisitos 6](#_Toc450378097)

[3. Criterios de comparación en la implementación 6](#_Toc450378098)

[3.1. Criterio 1: Horas empleadas en el desarrollo del sistema 6](#_Toc450378099)

[3.2. Criterio 2: Curva de aprendizaje 6](#_Toc450378100)

[3.3. Criterio 3: Herramientas externas 7](#_Toc450378101)

[3.4. Criterio 4: Estructura del código 7](#_Toc450378102)

[3.5. Criterio 5: Capacidad de adaptación 7](#_Toc450378103)

[3.6. Criterio 6: Rendimiento 7](#_Toc450378104)

[3.7. Criterio 7: Recursos existentes en la red 7](#_Toc450378105)

[4. Proyecto de implementación de un prototipo del sistema utilizando Ionic 8](#_Toc450378106)

[4.1. Documentación de diseño 8](#_Toc450378107)

[4.2. Documentación de construcción 8](#_Toc450378108)

[4.3. Documentación de pruebas 8](#_Toc450378109)

[4.3.1. Caso de prueba 1…n *(repetir este apartado, con sus subapartados, para los diferentes casos de prueba que se definan. Ojo! Todos los requisitos definidos deberían quedar verificados en alguno de los casos de prueba)* 8](#_Toc450378110)

[4.3.1.1. Requisitos a verificar en el caso de prueba 1 8](#_Toc450378111)

[4.3.1.2. Descripción de la prueba 1 8](#_Toc450378112)

[4.4. Incidencias detectadas en las pruebas 9](#_Toc450378113)

[4.5. Documentación de instalación 9](#_Toc450378114)

[4.6. 4.5 Manual de usuario 9](#_Toc450378115)

[5. Proyecto de implementación de un prototipo del sistema utilizando Apache Cordova 10](#_Toc450378116)

[5.1. Documentación de diseño 10](#_Toc450378117)

[5.2. Documentación de construcción 10](#_Toc450378118)

[5.3. Documentación de pruebas 10](#_Toc450378119)

[5.3.1. Caso de prueba 1…n 10](#_Toc450378120)

[5.3.1.1. Requisitos a verificar en el caso de prueba 1 10](#_Toc450378121)

[5.3.1.2. Descripción de la prueba 1 10](#_Toc450378122)

[5.4. Incidencias detectadas en las pruebas 10](#_Toc450378123)

[5.5. Documentación de instalación 11](#_Toc450378124)

[5.6. Manual de usuario 11](#_Toc450378125)

[6. Comparación de las dos implementaciones 12](#_Toc450378126)

[6.1. Evaluación de los criterios en la implementación usando Ionic 12](#_Toc450378127)

[6.2. Evaluación de los criterios en la implementación usando Apache Cordova 12](#_Toc450378128)

[7. Comparación de la implementación de las tecnologías 13](#_Toc450378129)

[8. Conclusiones 15](#_Toc450378130)

# Autores del trabajo, planificación y entrega

## Autores

El grupo que desarrolla el trabajo que se presenta a continuación el es grupo 7 de laboratorio del turno de tarde; este grupo está formado por:

* Miguel Ángel Rodríguez Blanco (Coordinador del grupo).
* Luis Nueda García.
* Álvaro Laza Martín.
* Eduardo Dorado Pérez.

## Planificación

La planificación de proyecto ha sido realizada por el Coordinador del grupo utilizando la herramienta de diagramación GanttPro, tras una reunión con el grupo de trabajo en la que se han expuesto las tareas necesarias para la consecución del objetivo final y se ha realizado un reparto/asignación de tareas equitativo.

Este trabajo de planificación está disponible en la URL:

***https://github.com/MiguelRodriguezBlanco/TG3/blob/0bfdcf1a8894c41750958ea6ffbba2fedb13afa8/Planificaci%C3%B3n%20TG3.pdf***

## Entrega

El repositorio creado para la puesta en común de los avances del grupo y en el que se ha guardado el resultado de este trabajo es:

***https://github.com/MiguelRodriguezBlanco/TG3***

En este repositorio se encuentra, además de los archivos subidos a los largo del desarrollo del trabajo los archivos requeridos en la rama máster:

* Informe del trabajo: con el nombre TG3\_final.docx
* Presentación del trabajo: TG3\_final.pptx
* Prototipos obtenidos implementando cada una de las tecnologías:
  + Prototipo Ionic\_final.zip
  + Prototipo Cordova\_final.zip

# Restaurante Multiplataforma

Es una solución completa que se presenta como una aplicación móvil, y un potente back-end que permite actualizaciones de datos/contenido.

La aplicación se adapta a las necesidades de cualquier restaurante que quiera presenta su carta a sus clientes, proporcionando una interfaz de usuario sencilla, que permite a navegar a través de todas las categorías del menú, visualizar platos de la carta…La aplicación a desarrollar presentará un paquete de características de gran utilidad adaptadas a las necesidades y tecnologías de hoy en día, tales como compartir información a través de las redes sociales, informar sobre ofertas especiales y videos de YouTube.

Principales Características:

* Solo se necesitan conocimientos de HTML y CSS.
* Interfaz de usuario sencilla, limpia e intuitiva.
* Arquitectura modular que permite adaptación a las necesidades de cada cliente.
* Dispone de una amplia gama de colores depurada
* Instalación guiada paso a paso, rápida e intuitiva.
* Mantenimiento y actualización de contenidos sencillo
* Actualizaciones gratuitas: aplicación en constante evolución, surgen constantemente nuevas versiones depuradas y que aplican sus capacidades.
* Una base de código multiplataforma, compatible con Android y iOS.
* Soporte técnico rápido y eficiente.

## Requisitos funcionales

En la siguiente tabla se presentan los requisitos funcionales del prototipo

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| RF01 | Aplicación multiplataforma |
| RF02 | Desarrollo sobre JavaScript, CSS, HTML |
| RF03 | Dispondrá de Plantillas sobre la que presentar la carta |
| RF04 | Dispondrá de Plantillas con información sobre las ofertas especiales |
| RF05 | Aplicación fácil de usar |
| RF06 | Instalación paso a paso, guiada y sencilla |
| RF07 | Permitirá visualizar imágenes asociadas a cada uno de los platos de la carta. |
| RF08 | Desde la pantalla principal será posible acceder a las diferentes categorías de menú |
| RF09 | Desde cada submenú se puede avanzar hacia delante, atrás o acceder a la pantalla principal |
| RF10 | Será posible acceder a redes sociales a través de la app. |

## Otros requisitos

En la siguiente tabla se presenta el catálogo de requisitos no funcionales del sistema.

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| ***RI*** | ***Requisitos de Interfaz*** |
| RI01 | Interfaz sencilla e intuitiva |
| RI02 | Interfaz en color |
| RI03 | Visualización de la carta en color |
| ***RR*** | ***Requisitos de Rendimiento*** |
| R01 | Tiempos de respuesta rápidos |
| ***RSw*** | ***Requisitos Sw*** |
| RSw01 | Arquitectura modular |
| RSw02 | Funcionará al menos sobre los siguientes sistemas operativos:   * + Windows 8   + OsX   + Android   + iOS |
| ***RHw*** | ***Requisitos Hw*** |
| RHw01 | 256MB RAM mínima requerida para el dispositivo sobre el que se instala la app |
| RHw02 | 1 GB disco duro mínimo requerido para el dispositivo sobre el que se instala la app |
| RHw03 | Espacio de disco: 160 MB Libres |
| RHw04 | Mínimo un Pentium 4, a 1 GHz |
| RHw05 | Doble núcleo 2GHz |

# Criterios de comparación en la implementación

## Criterio 1: Horas empleadas en el desarrollo del sistema

**Descripción:** Tiempo requerido para el desarrollo de la aplicación.

**Tipo de valor:** Numérico (horas).

## Criterio 2: Curva de aprendizaje

**Descripción:** Promedio de tiempo que necesita un usuario para conocer las utilidades básicas de la aplicación.

**Tipo de valor:** Numérico (horas).

## Criterio 3: Herramientas externas

**Descripción:** Compatibilidad con el uso de herramientas externas.

**Tipo de valor:** Booleano (Si/No).

## Criterio 4: Estructura del código

**Descripción:** Número de paquetes que forman el código de la aplicación

**Tipo de valor:** Numérico.

## Criterio 5: Capacidad de adaptación

**Descripción:** Nivel de complejidad que presenta la aplicación para al usuario final a la hora de incorporar a la aplicación toda la información relativa su restaurante: menús, ofertas, imágenes….

**Tipo de valor:** Escala: 1…5 (1: muy fácil…5: muy complejo).

## Criterio 6: Rendimiento

**Descripción:** Tiempo de respuesta medio al ejecutar una utilidad de la aplicación

**Tipo de valor:** Numérico (segundos).

## Criterio 7: Tiempo de arranque

**Descripción:** Tiempo que tarda en arrancar la aplicación

**Tipo de valor:** Numérico (segundos)

## Criterio 8: Recursos existentes en la red

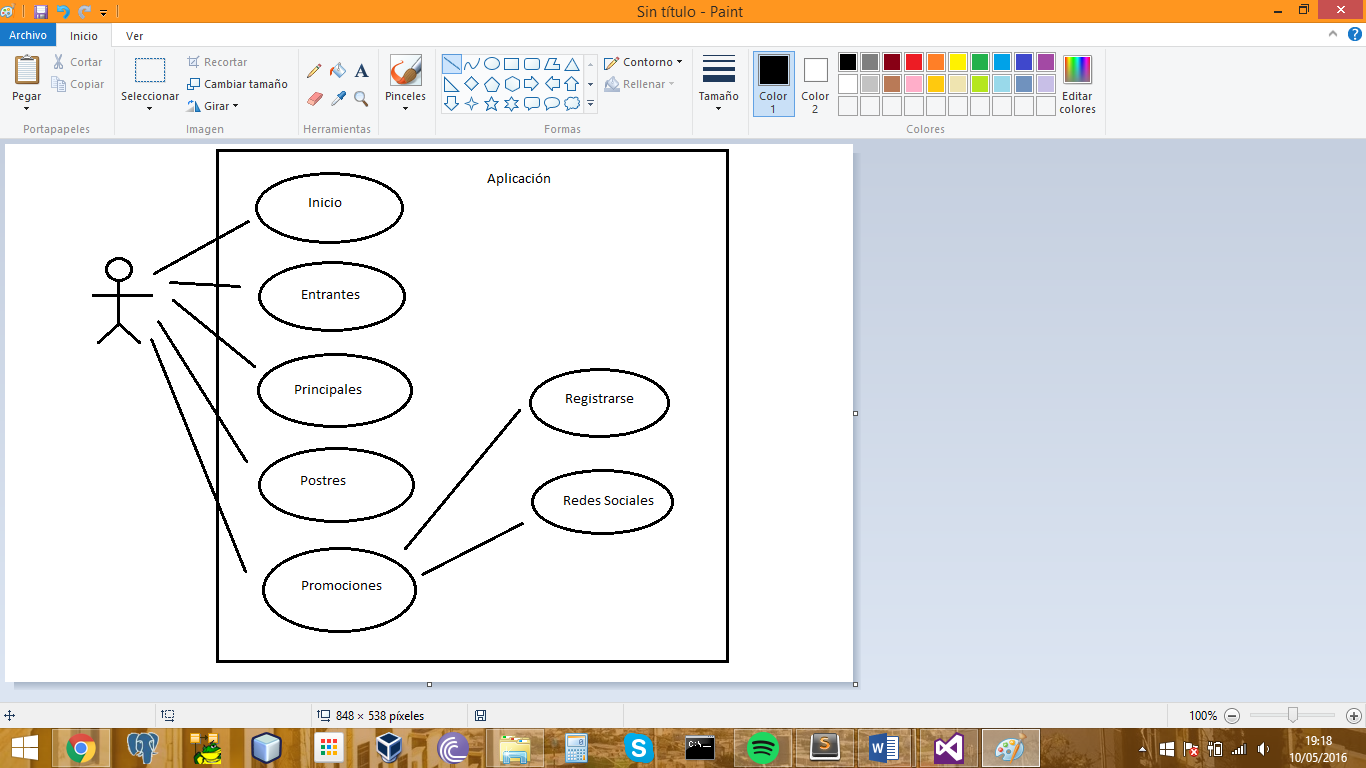
**Descripción:** Información existente en la red de soporte para el desarrollo de la aplicación.

**Tipo de valor:** Escala: escasa, media, abundante.

# Proyecto de implementación de un prototipo del sistema utilizando Ionic

## Documentación de diseño

Previo al desarrollo de la aplicación se plantea el caso de uso que presenta las interacciones necesarias entre el usuario y los diferentes submenús y uno de estos y las diferentes funcionalidades que incluye. Todas estas interacciones quedan identificadas y representadas como se indica en el caso de uso que se muestra a continuación.



A partir del caso de uso se desarrolla la aplicación que como se presenta al usuario a través de una pantalla principal un “Sidemenu”, con una SideBar, con diferentes pestañas.



## Documentación de construcción

El código de la aplicación se agrupa en diferentes paquetes, la estructura de paquetes que lo forman estos paquetes se presenta a continuación:



Dentro de esta estructura, los principales paquetes, aquellos que dan entidad a la aplicación se presentan son los que detallan a continuación:

* **img**: carpeta en la que se almacenan las imágenes que la aplicación incorpora asociadas a los platos del menú o las promociones.
* **templates**: plantillas html que dan formato a todas las categorías del menú de la aplicación.
* **js**: carpeta en la que se almacena el código Java Script.
* **css**: diferentes estilos que utiliza la aplicación.

## Documentación de pruebas

## Caso de prueba 1: Sintaxis del código.

## Requisitos a verificar en el caso del contenido.

A través de este caso de prueba se verificará el cumplimiento de los siguientes requisitos.

|  |  |
| --- | --- |
| RF01 | Aplicación multiplataforma |
| RF02 | Desarrollo sobre JavaScript, CSS, HTML |
| RF03 | Dispondrá de Plantillas sobre la que presentar la carta |
| RF04 | Dispondrá de Plantillas con información sobre las ofertas especiales |
| RSw01 | Arquitectura modular |

## Descripción de la prueba: Sintaxis del código

Este caso de prueba trata de verificar la sintaxis del código desarrollado recorriéndolo con el entorno de desarrollo, de modo que se demuestra que el código es claro y se ha definido correctamente.

| **Descripción** | **Resultado esperado** | **Resultado obtenido (Ok/Nok)** |
| --- | --- | --- |
| Recorrer el código con el entorno de desarrollo | Mostrar el código tabulado y bien estructurado, identificando por colores los diferentes clases, atributos etc. | **Ok** |

## Caso de prueba 2: Funcionalidad e interfaz de la aplicación.

## Requisitos a verificar en el caso del contenido.

A través de este caso de prueba se verificará el cumplimiento de los siguientes requisitos.

|  |  |
| --- | --- |
| RF05 | Aplicación sencilla, fácil de usar |
| RI01 | Interfaz sencilla e intuitiva |
| RI02 | Interfaz en color |
| RI03 | Visualización de la carta en color |

## Descripción de la prueba: Funcionalidad e interfaz de la aplicación.

A través de la ejecución de este caso de prueba, se trata de verificar la funcionalidad de la aplicación, así como la interfaz de usuario.

| **Descripción** | **Resultado esperado** | **Resultado obtenido (Ok/Nok)** |
| --- | --- | --- |
| Testeamos la aplicación desde ionic serve. | Ejecuta la aplicación con el diseño establecido. | **Ok** |

## Caso de prueba 3: Navegación.

## Requisitos a verificar en el caso del contenido.

A través de este caso de prueba se verificará el cumplimiento de los siguientes requisitos.

|  |  |
| --- | --- |
| RF08 | Desde la pantalla principal será posible acceder a las diferentes categorías de menú |
| RF09 | Desde cada submenú se puede avanzar hacia delante, atrás o acceder a la pantalla principal |
| R01 | Tiempos de respuesta rápidos |

## Descripción de la prueba: Navegación

Aplica casos de uso en el diseño de casos de prueba que ejerciten cada escenario de uso contra el diseño de navegación.

| **Descripción** | **Resultado esperado** | **Resultado obtenido (Ok/Nok)** |
| --- | --- | --- |
| Se navega a través de los diferentes menús disponibles en la aplicación, yendo hacia delante hacia detrás, saltando de uno a otro. | Es posible acceder a todos los menús y submenús disponibles, avanzando hacia delante y hacia atrás. | **Ok** |

## Caso de prueba 4: Componentes.

## Requisitos a verificar en el caso del contenido.

A través de este caso de prueba se verificará el cumplimiento de los siguientes requisitos.

|  |  |
| --- | --- |
| RF10 | Será posible acceder a redes sociales a través de la app. |
| RI01 | Interfaz sencilla e intuitiva |
| RI02 | Interfaz en color |
| RI03 | Visualización de la carta en color |

## Descripción de la prueba: Componentes

Ejercita el contenido y las unidades funcionales dentro de la web app, cada elemento de la arquitectura de la web app se prueba de manera unitaria en la medida de lo posible.

| **Descripción** | **Resultado esperado** | **Resultado obtenido (Ok/Nok)** |
| --- | --- | --- |
| Es posible acceder desde la aplicación a las aplicaciones de redes sociales disponibles en el dispositivo móvil así como publicar contenido de la app en ellas. | Desde la app es posible acceder a y visualizar correctamente el contenido de:   * Facebook, * Youtube | **Ok** |

## Caso de prueba 5: Configuración.

## Requisitos a verificar en el caso del contenido.

A través de este caso de prueba se verificará el cumplimiento de los siguientes requisitos.

|  |  |
| --- | --- |
| RF05 | Aplicación sencilla, fácil de usar |
| RF07 | Permitirá visualizar imágenes asociadas a cada uno de los platos de la carta. |
| RF08 | Desde la pantalla principal será posible acceder a las diferentes categorías de menú |
| RF09 | Desde cada submenú se puede avanzar hacia delante, atrás o acceder a la pantalla principal |
| RF10 | Será posible acceder a redes sociales a través de la app. |
| RI01 | Interfaz sencilla e intuitiva |
| RI02 | Interfaz en color |
| RI03 | Visualización de la carta en color |

## Descripción de la prueba: Configuración

Intentan descubrir errores específicos respecto de un determinado cliente o ambiente de servidores particulares. Las pruebas se encaminan a descubrir los errores asociados con cada posible configuración.

| **Descripción** | **Resultado esperado** | **Resultado obtenido (Ok/Nok)** |
| --- | --- | --- |
| Se instala la app para una configuración preestablecida se navega por ella verificando que se comporta y se visualiza el contenido correctamente | La app se instala de forma rápida y sencilla. Una vez instalada, cada pantalla de la app se visualiza correctamente permitiendo navegar por los diferentes menús disponibles sin errores. | **Ok** |

## Caso de prueba 6: Multiplataforma.

## Requisitos a verificar en el caso del contenido.

A través de este caso de prueba se verificará el cumplimiento de los siguientes requisitos.

|  |  |
| --- | --- |
| RF05 | Aplicación sencilla, fácil de usar |
| RF06 | Instalación paso a paso, guiada y sencilla |
| RF07 | Permitirá visualizar imágenes asociadas a cada uno de los platos de la carta. |
| RF08 | Desde la pantalla principal será posible acceder a las diferentes categorías de menú |
| RF09 | Desde cada submenú se puede avanzar hacia delante, atrás o acceder a la pantalla principal |
| RF10 | Será posible acceder a redes sociales a través de la app. |
| RI01 | Interfaz sencilla e intuitiva |
| RI02 | Interfaz en color |
| RI03 | Visualización de la carta en color |
| RSw02 | Funcionará al menos sobre los siguientes sistemas operativos:   * + Windows 8   + OsX   + Android   + iOS |
| RHw01 | 256MB RAM mínima requerida para el dispositivo sobre el que se instala la app |
| RHw02 | 1 GB disco duro mínimo requerido para el dispositivo sobre el que se instala la app |
| RHw03 | Espacio de disco: 160 MB Libres |
| RHw04 | Mínimo un Pentium 4, a 1 GHz |
| RHw05 | Doble núcleo 2GHz |

## Descripción de la prueba: configuración

Se verifica la característica multiplataforma de la app.

| **Descripción** | **Resultado esperado** | **Resultado obtenido (Ok/Nok)** |
| --- | --- | --- |
| Se instala la app en móviles con diferentes sistemas operativos con las siguientes características mínimas:   * 256MB RAM * 1 GB disco duro * 160 MB Libres * Pentium 4, a 1 GHz * Doble núcleo 2GHz | La app se instala de forma rápida y sencilla correctamente. Una vez instalada, se verifica el funcionamiento correcto. | **Ok** |

## Incidencias detectadas en las pruebas

|  |  |  |
| --- | --- | --- |
| **Caso de prueba** | **Incidencia** | **Acciones correctoras** |
|  |  |  |
|  |  |  |
|  |  |  |

## Documentación de instalación

*Descripción suficiente para que una persona que no ha participado en el proyecto pueda instalar el prototipo.*

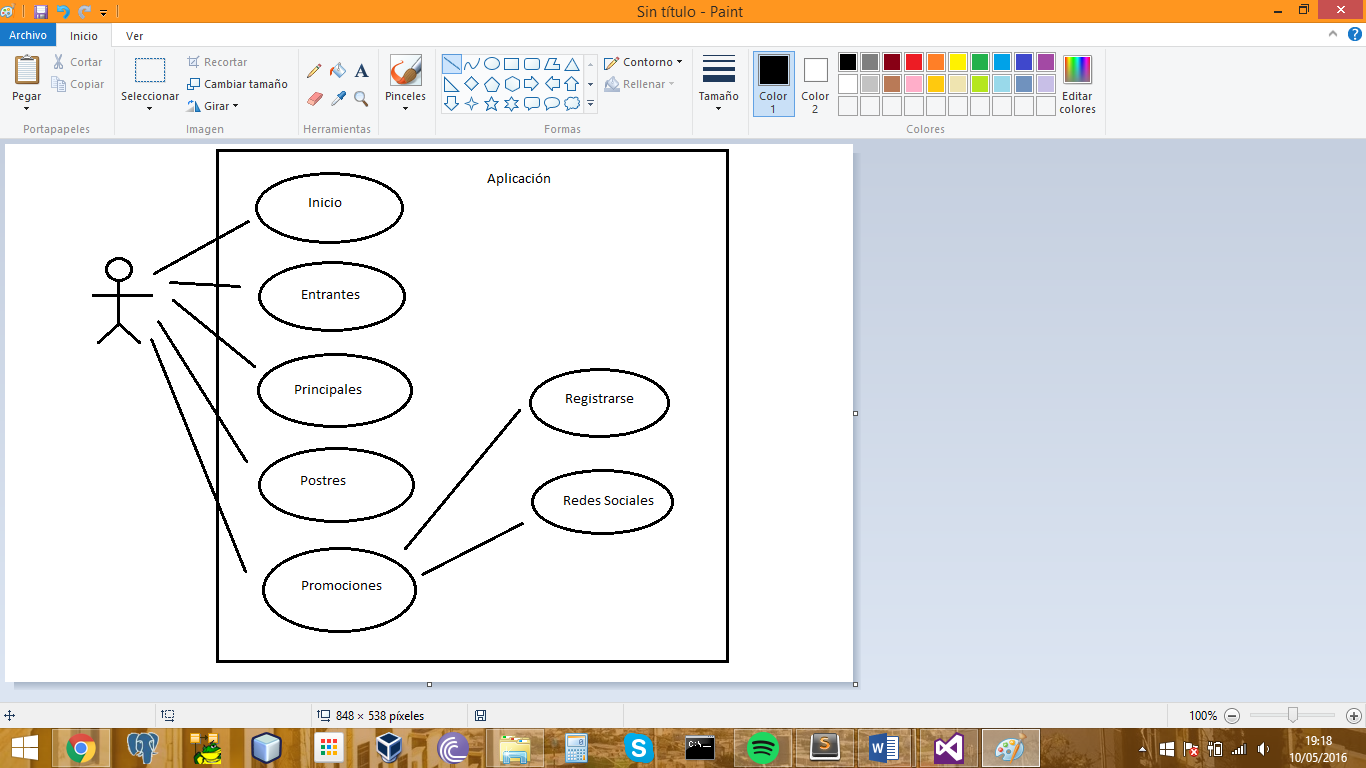
## 4.5 Manual de usuario

*Descripción suficiente para que una persona que no ha participado en el proyecto pueda utilizar toda la funcionalidad que ofrece el prototipo. Que debe coincidir con los requisitos funcionales incluidos en el apartado 2.*

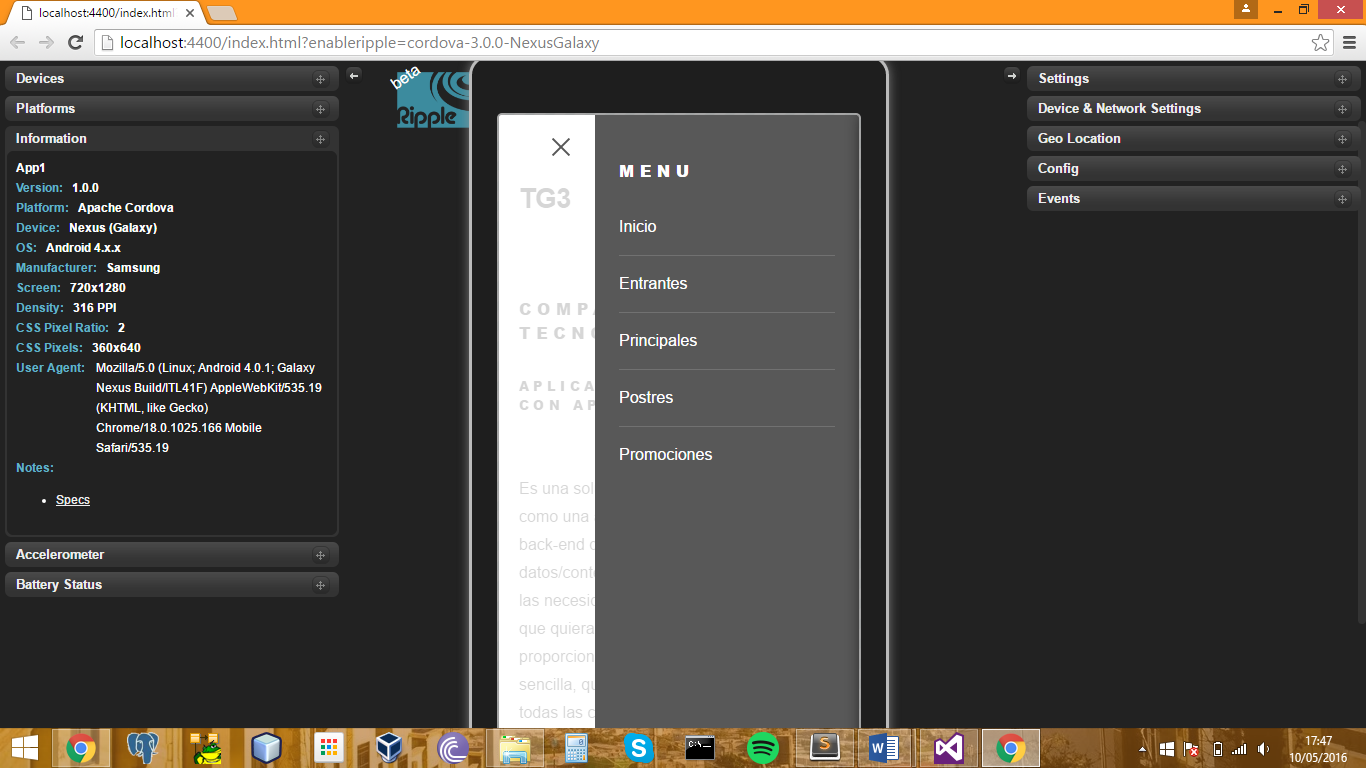
# Proyecto de implementación de un prototipo del sistema utilizando Apache Cordova

## Documentación de diseño

Previo al desarrollo de la aplicación se plantea el caso de uso que presenta las interacciones necesarias entre el usuario y los diferentes submenús y uno de estos y las diferentes funcionalidades que incluye. Todas estas interacciones quedan identificadas y representadas como se indica en el caso de uso que se muestra a continuación.

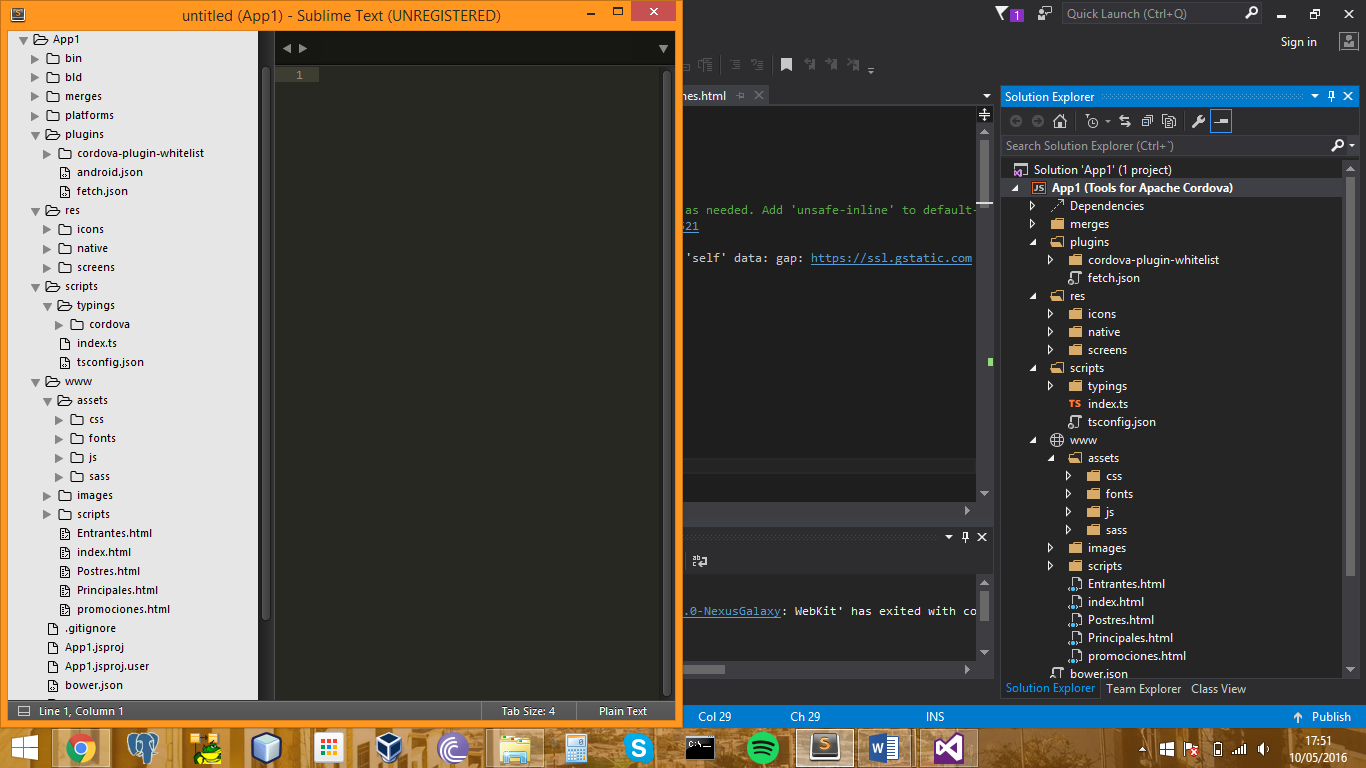


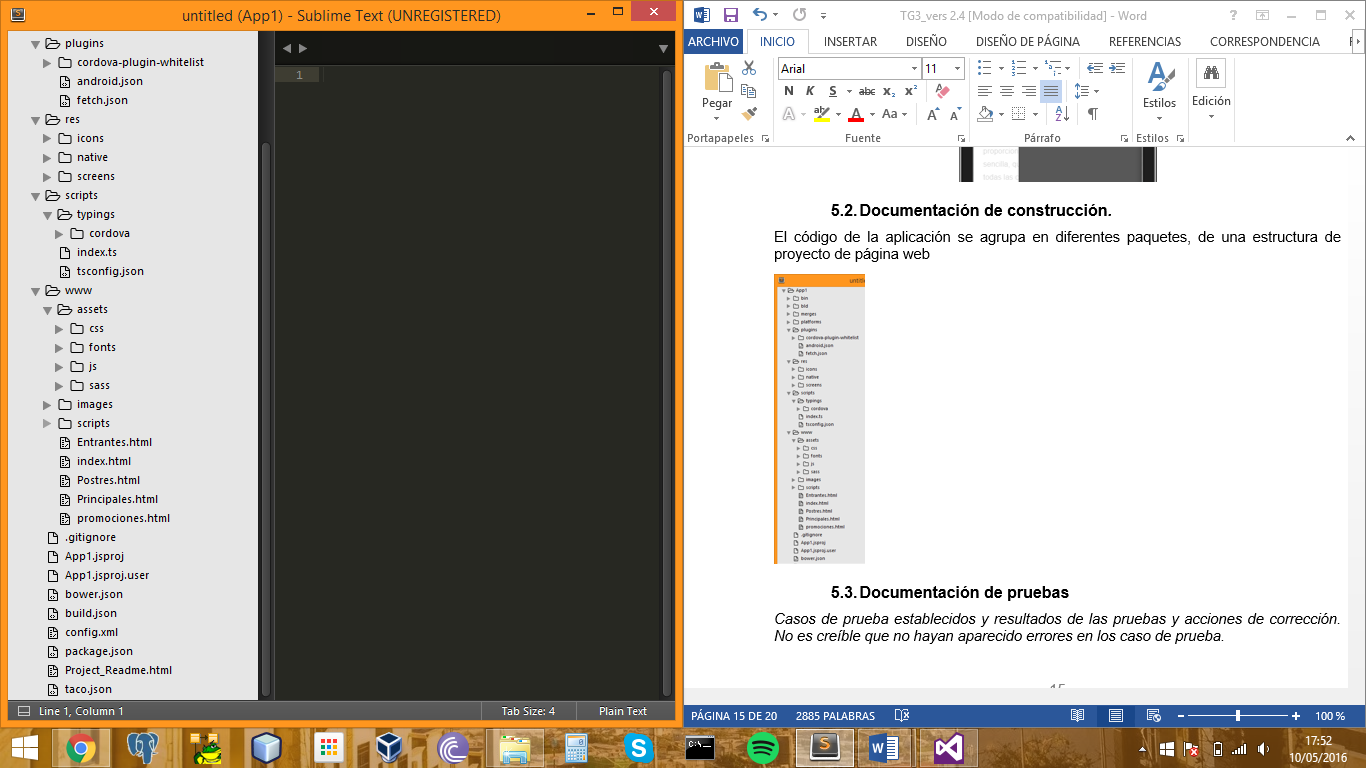
A partir del caso de uso se desarrolla la aplicación que como se presenta al usuario a través de una pantalla principal un “Sidemenu”, con una SideBar, con diferentes pestañas.



## Documentación de construcción*.*

El código de la aplicación se agrupa en diferentes paquetes, la estructura de paquetes que lo forman el proyecto de página web se presenta a continuación:





* **plugins:** Carpeta que contiene la información y los plugins de apache cordova que han sido instalados para la aplicación
* **res:** Archivos de configuración de apache cordova para las distintas plataformas (android, ios, wp). Contiene elementos gráficos y formato de dispositivos para su posterior emulación.
* **scripts:** Librería que contiene clases de los distintos plugins de apache cordova que pueden ser usados en nuestro proyecto
* **www:** Carpeta “principal” de nuestra aplicación.
  + **assets:** Contiene los archivos que sirven para dar estilo a nuestra aplicación (css; hoja de estilos, fonts; documento para iconos, js; contiene código JavaScript, sass; directorio que se complementa con css)
  + **images:** Incorpora las imágenes que usaremos en nuestra aplicación
  + **scripts:** Archivos de arranque de la aplicación. Función onDeviceReady

## Documentación de pruebas

## Caso de prueba 1: Sintaxis del código

## Requisitos a verificar en el caso de prueba del contenido

A través de este caso de prueba se verificará el cumplimiento de los siguientes requisitos.

|  |  |
| --- | --- |
| RF01 | Aplicación multiplataforma |
| RF02 | Desarrollo sobre JavaScript, CSS, HTML |
| RF03 | Dispondrá de Plantillas sobre la que presentar la carta |
| RF04 | Dispondrá de Plantillas con información sobre las ofertas especiales |
| RSw01 | Arquitectura modular |

## Descripción de la prueba: Sintaxis del código

Este caso de prueba trata de verificar la sintaxis del código desarrollado recorriéndolo con el entorno de desarrollo, de modo que se demuestra que el código es claro y se ha definido correctamente.

| **Descripción** | **Resultado esperado** | **Resultado obtenido (Ok/Nok)** |
| --- | --- | --- |
| Recorrer el código con el entorno de desarrollo | Mostrar el código tabulado y bien estructurado, identificando por colores los diferentes clases, atributos etc. | **Ok** |

## Caso de prueba 2: Funcionalidad e interfaz de la aplicación

## Requisitos a verificar en el caso del contenido

A través de este caso de prueba se verificará el cumplimiento de los siguientes requisitos

|  |  |
| --- | --- |
| RF05 | Aplicación sencilla, fácil de usar |
| RI01 | Interfaz sencilla e intuitiva |
| RI02 | Interfaz en color |
| RI03 | Visualización de la carta en color |

## Descripción de la prueba: Funcionalidad e interfaz de la aplicación

A través de la ejecución de este caso de prueba, se trata de verificar la funcionalidad de la aplicación, así como la interfaz de usuario.

| **Descripción** | **Resultado esperado** | **Resultado obtenido (Ok/Nok)** |
| --- | --- | --- |
| Testeamos la aplicación desde el emulador de Visual Studio de Apache Cordova. | Ejecuta la aplicación con el diseño establecido. | **Ok** |

## Caso de prueba 3: Navegación

## Requisitos a verificar en el caso del contenido

A través de este caso de prueba se verificará el cumplimiento de los siguientes requisitos.

|  |  |
| --- | --- |
| RF08 | Desde la pantalla principal será posible acceder a las diferentes categorías de menú |
| RF09 | Desde cada submenú se puede avanzar hacia delante, atrás o acceder a la pantalla principal |
| R01 | Tiempos de respuesta rápidos |

## Descripción de la prueba: Navegación

Aplica casos de uso en el diseño de casos de prueba que ejerciten cada escenario de uso contra el diseño de navegación.

| **Descripción** | **Resultado esperado** | **Resultado obtenido (Ok/Nok)** |
| --- | --- | --- |
| Se navega a través de los diferentes menús disponibles en la aplicación, yendo hacia delante hacia detrás, saltando de uno a otro. | Es posible acceder a todos los menús y submenús disponibles, avanzando hacia delante y hacia atrás. | **Ok** |

## Caso de prueba 4: Componentes

## Requisitos a verificar en el caso del contenido

A través de este caso de prueba se verificará el cumplimiento de los siguientes requisitos.

|  |  |
| --- | --- |
| RF10 | Será posible acceder a redes sociales a través de la app. |
| RI01 | Interfaz sencilla e intuitiva |
| RI02 | Interfaz en color |
| RI03 | Visualización de la carta en color |

## Descripción de la prueba: Componentes

Ejercita el contenido y las unidades funcionales dentro de la web app, cada elemento de la arquitectura de la web app se prueba de manera unitaria en la medida de lo posible.

| **Descripción** | **Resultado esperado** | **Resultado obtenido (Ok/Nok)** |
| --- | --- | --- |
| Es posible acceder desde la aplicación a las aplicaciones de redes sociales disponibles en el dispositivo móvil así como publicar contenido de la app en ellas. | Desde la app es posible acceder a y visualizar correctamente el contenido de:   * Facebook, * Youtube | **Ok** |

## Caso de prueba 5: Configuración

## Requisitos a verificar en el caso del contenido

A través de este caso de prueba se verificará el cumplimiento de los siguientes requisitos.

|  |  |
| --- | --- |
| RF05 | Aplicación sencilla, fácil de usar |
| RF07 | Permitirá visualizar imágenes asociadas a cada uno de los platos de la carta. |
| RF08 | Desde la pantalla principal será posible acceder a las diferentes categorías de menú |
| RF09 | Desde cada submenú se puede avanzar hacia delante, atrás o acceder a la pantalla principal |
| RF10 | Será posible acceder a redes sociales a través de la app. |
| RI01 | Interfaz sencilla e intuitiva |
| RI02 | Interfaz en color |
| RI03 | Visualización de la carta en color |

## Descripción de la prueba: Configuración

Intentar descubrir errores específicos respecto de un determinado cliente o ambiente de servidores particulares. Las pruebas se encaminan a descubrir los errores asociados con cada posible configuración.

| **Descripción** | **Resultado esperado** | **Resultado obtenido (Ok/Nok)** |
| --- | --- | --- |
| Se instala la app para una configuración preestablecida se navega por ella verificando que se comporta y se visualiza el contenido correctamente | La app se instala de forma rápida y sencilla. Una vez instalada, cada pantalla de la app se visualiza correctamente permitiendo navegar por los diferentes menús disponibles sin errores. | **Ok** |

## Caso de prueba 6: Multiplataforma

## Requisitos a verificar en el caso del contenido

A través de este caso de prueba se verificará el cumplimiento de los siguientes requisitos.

|  |  |
| --- | --- |
| RF05 | Aplicación sencilla, fácil de usar |
| RF06 | Instalación paso a paso, guiada y sencilla |
| RF07 | Permitirá visualizar imágenes asociadas a cada uno de los platos de la carta. |
| RF08 | Desde la pantalla principal será posible acceder a las diferentes categorías de menú |
| RF09 | Desde cada submenú se puede avanzar hacia delante, atrás o acceder a la pantalla principal |
| RF10 | Será posible acceder a redes sociales a través de la app. |
| RI01 | Interfaz sencilla e intuitiva |
| RI02 | Interfaz en color |
| RI03 | Visualización de la carta en color |
| RSw02 | Funcionará al menos sobre los siguientes sistemas operativos:   * + Windows 8   + OsX   + Android   + iOS |
| RHw01 | 256MB RAM mínima requerida para el dispositivo sobre el que se instala la app |
| RHw02 | 1 GB disco duro mínimo requerido para el dispositivo sobre el que se instala la app |
| RHw03 | Espacio de disco: 160 MB Libres |
| RHw04 | Mínimo un Pentium 4, a 1 GHz |
| RHw05 | Doble núcleo 2GHz |

## Descripción de la prueba: configuración

Se verifica la característica multiplataforma de la app.

| **Descripción** | **Resultado esperado** | **Resultado obtenido (Ok/Nok)** |
| --- | --- | --- |
| Se instala la app en móviles con diferentes sistemas operativos con las siguientes características mínimas:   * 256MB RAM * 1 GB disco duro * 160 MB Libres * Pentium 4, a 1 GHz * Doble núcleo 2GHz | La app se instala de forma rápida y sencilla correctamente. Una vez instalada, se verifica el funcionamiento correcto. | **Ok** |

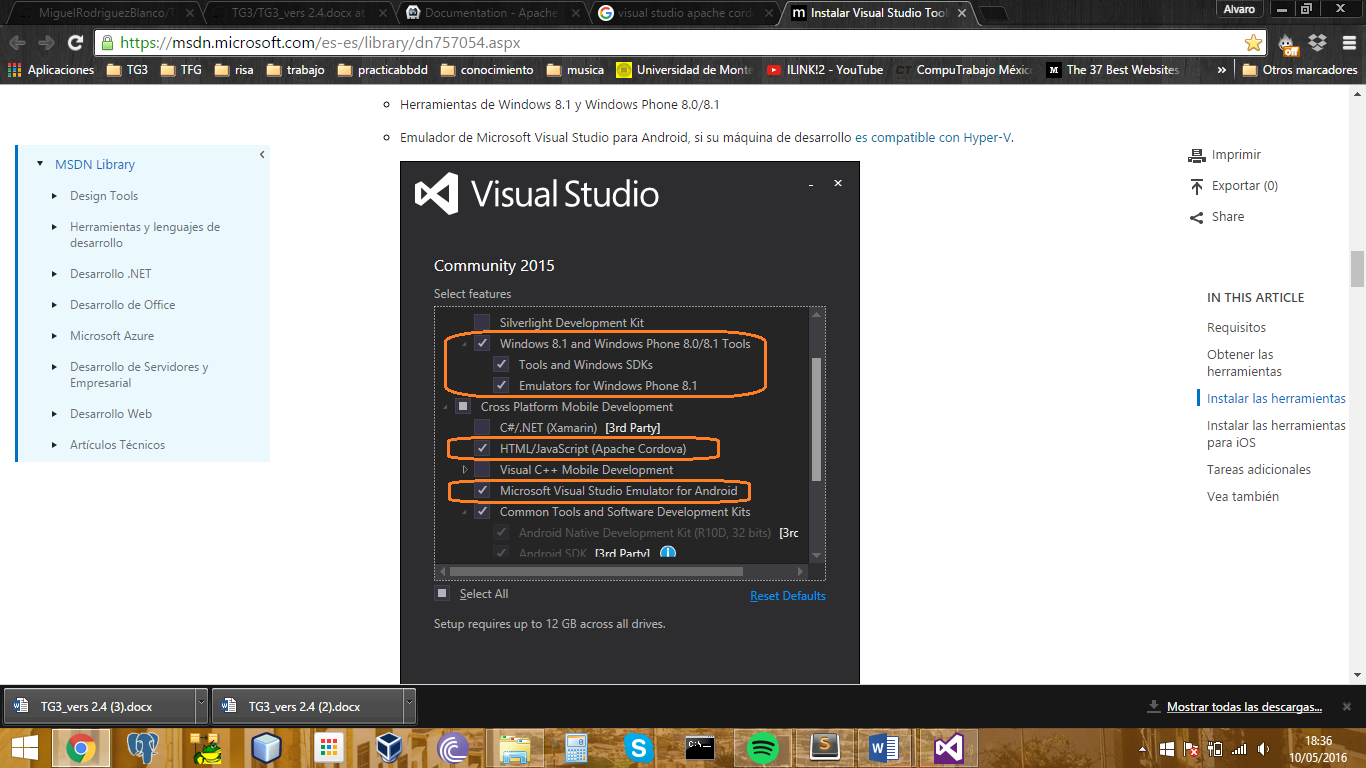
## Incidencias detectadas en las pruebas

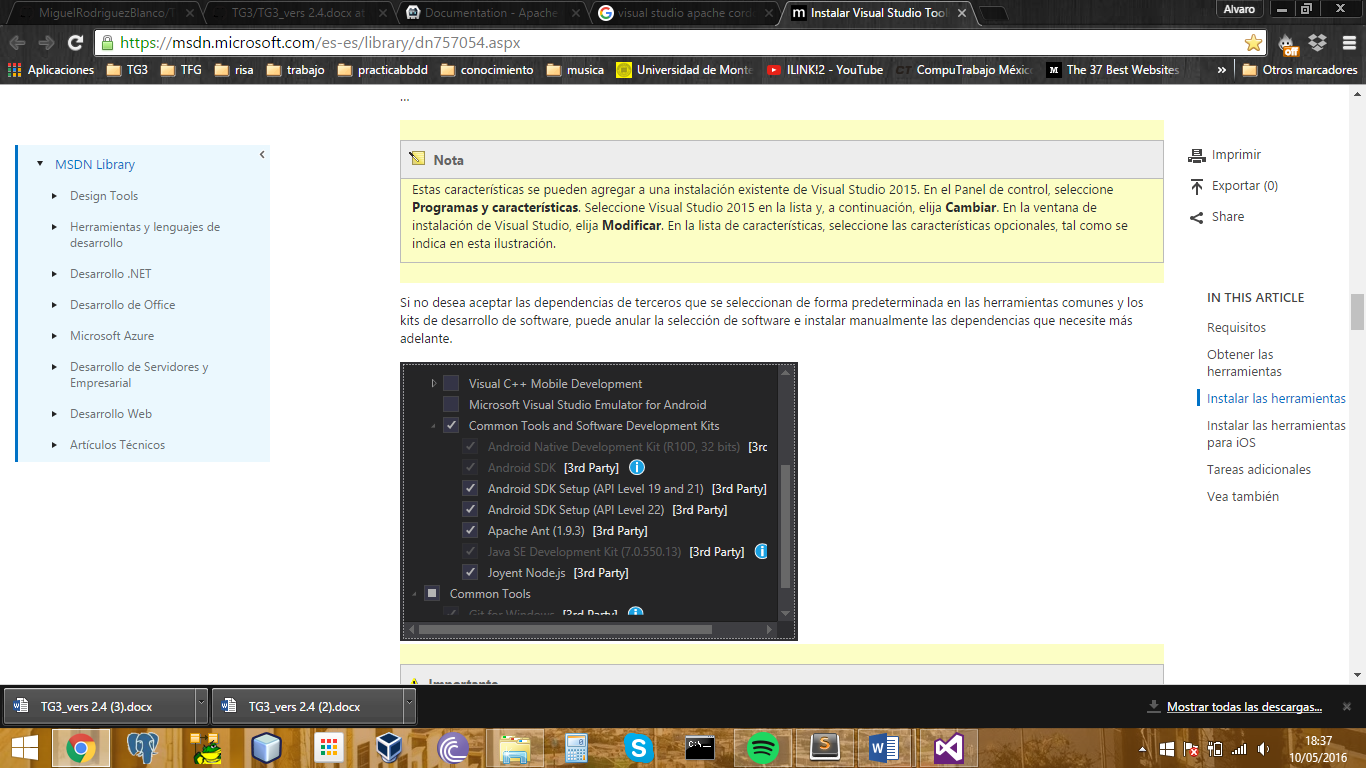
|  |  |  |
| --- | --- | --- |
| **Caso de prueba** | **Incidencia** | **Acciones correctoras** |
| **2** | Apache Cordova no cuenta con interfaz propia. A la hora de generar una, se encontraron problemas de dependencias | Creación de directorio de la interfaz incluyendo todos los paquetes necesarios (css, js y fonts) |
| **6** | A la hora de arrancar al app en un entorno virtual, se generaban errores de dependencias y paquetes mal instalados, imposibilitando la ejecución en la máquina virtual | Se procedió a desinstalar los elementos de Apache Cordova (apache ant, sdk android, node.js y apache), para volverlos a instalar dentro de la solución de Visual Studio |

## Documentación de instalación

Para la instalación y uso de Apache Cordova en un primer momento se decidió a instalar el software tal y como se explica en la documentación oficial de Apache Cordova (<https://cordova.apache.org/docs/en/latest/>). Durante el proceso de instalación no existieron problemas algunos, pero a la hora de depurar el programa o intentar emularlo en una imagen virtual nos encontramos con numerosos problemas como la falta de dependencias o la imposibilidad de conexión con dispositivos virtuales, por lo que se decidió instalar el complemento de desarrollo web y Apache Cordova para Visual Studio.

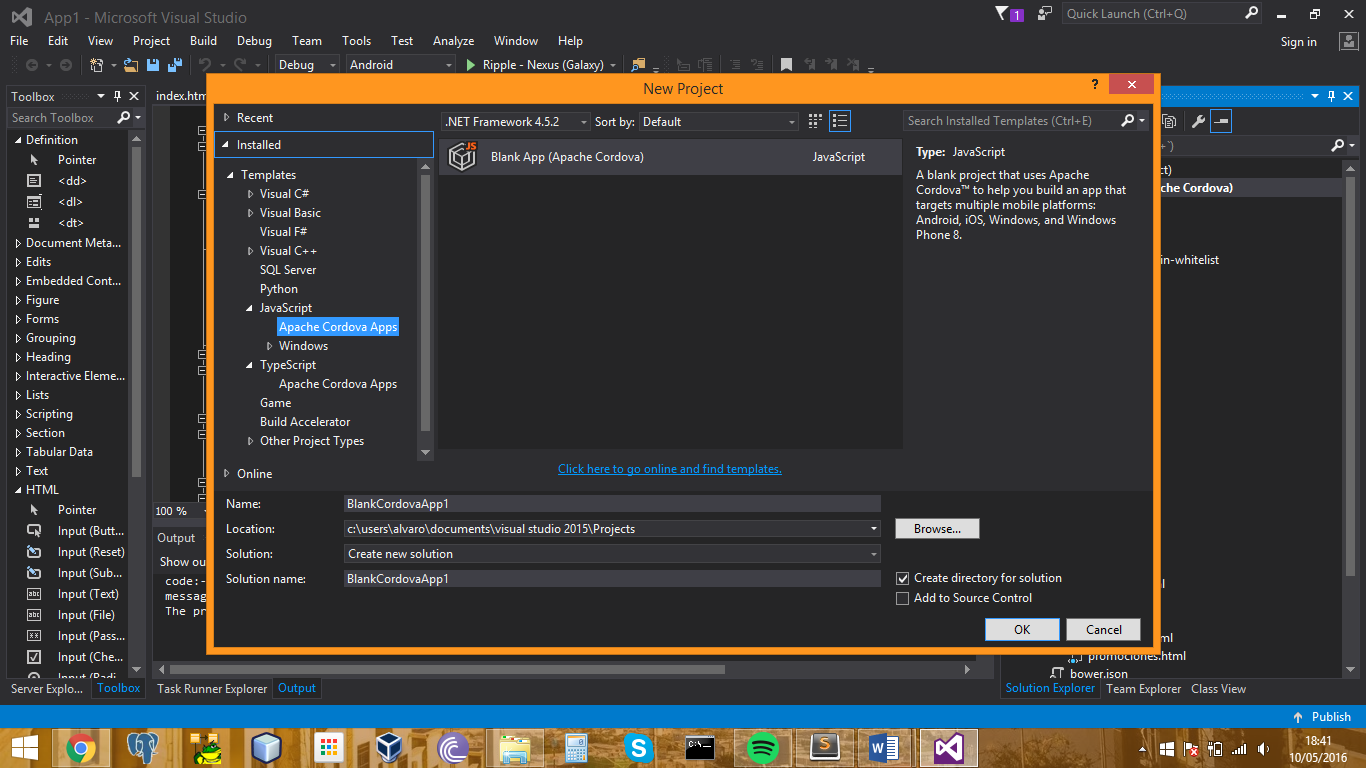
En el proceso de instalación de Visual Studio tenemos que incluir las siguientes características:



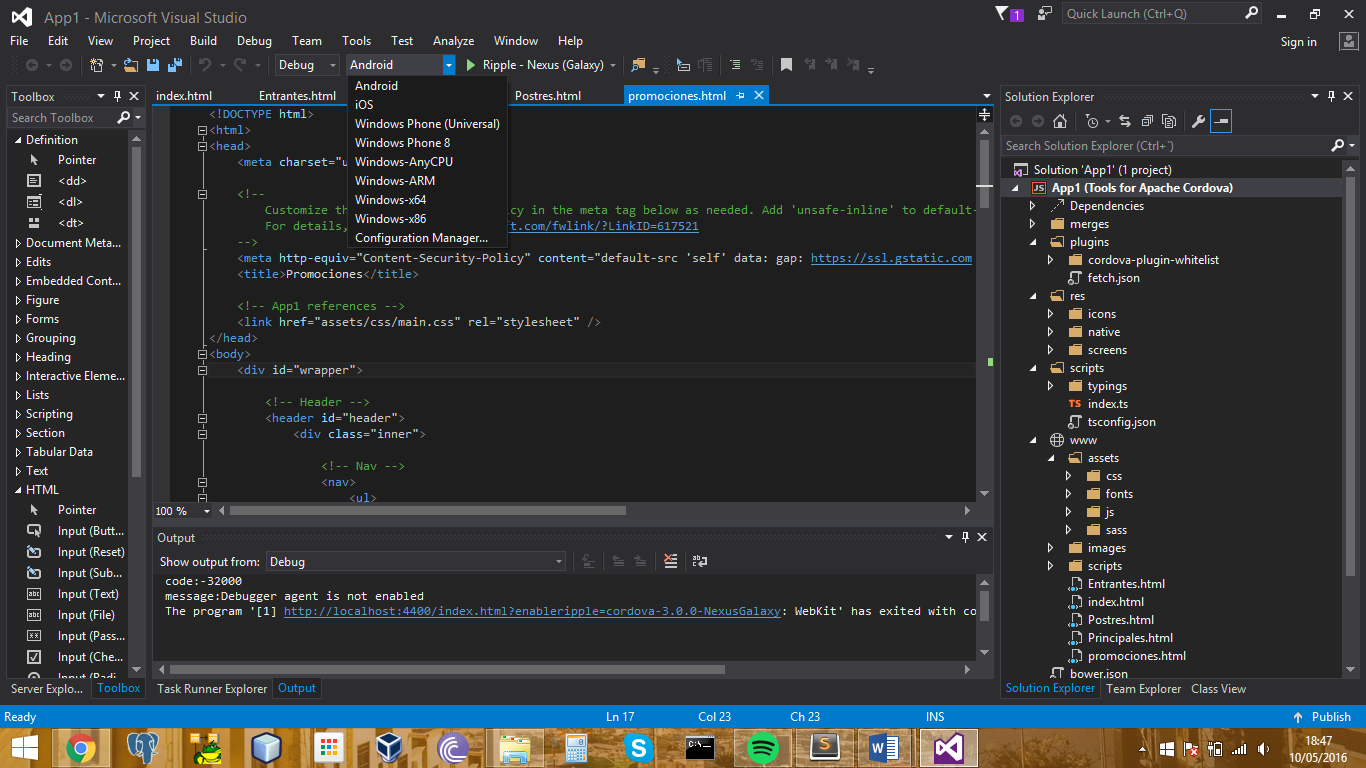


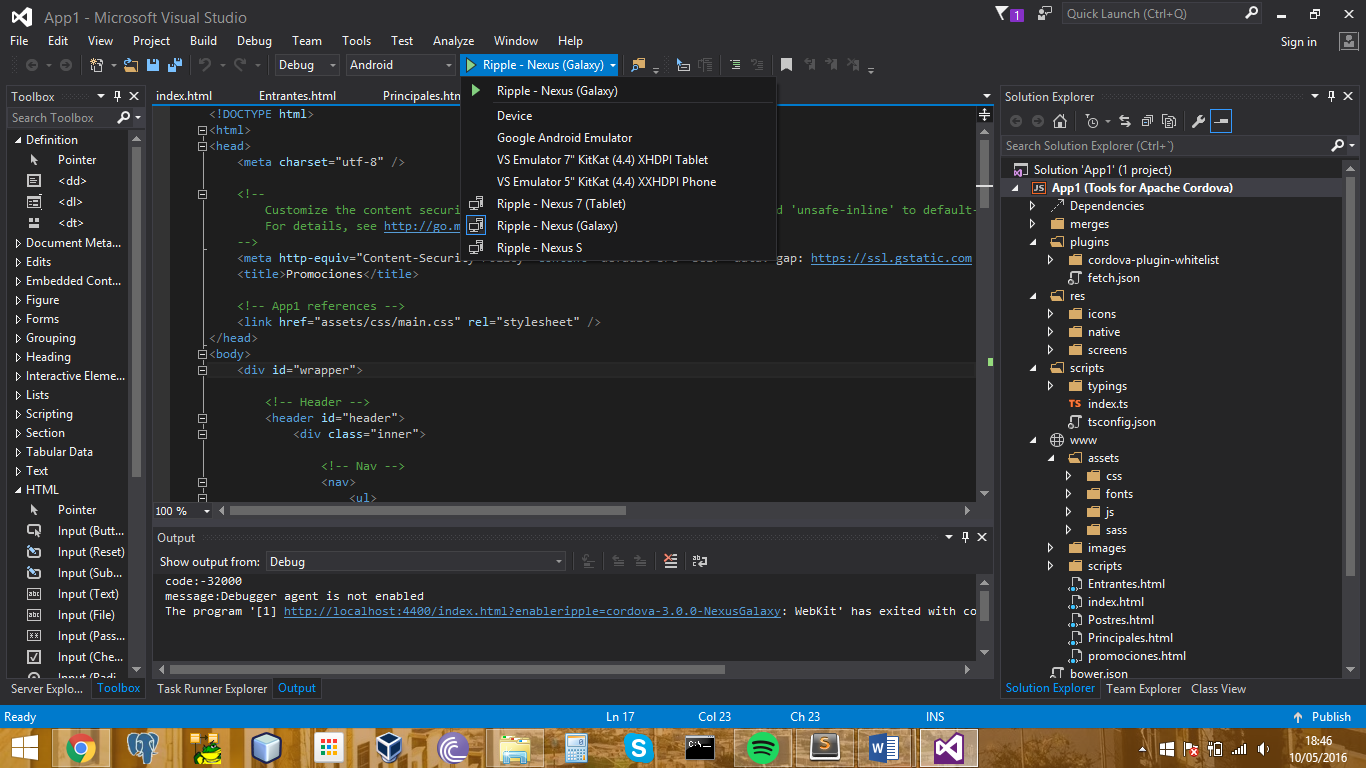
Al incluir todas las características anteriores, Visual Studio está instalando todos los complementos que necesitamos para la creación y emulación de proyectos para Apache Cordova, además de poder hacer uso de su interfaz que nos facilita el trabajo a la hora de programar.

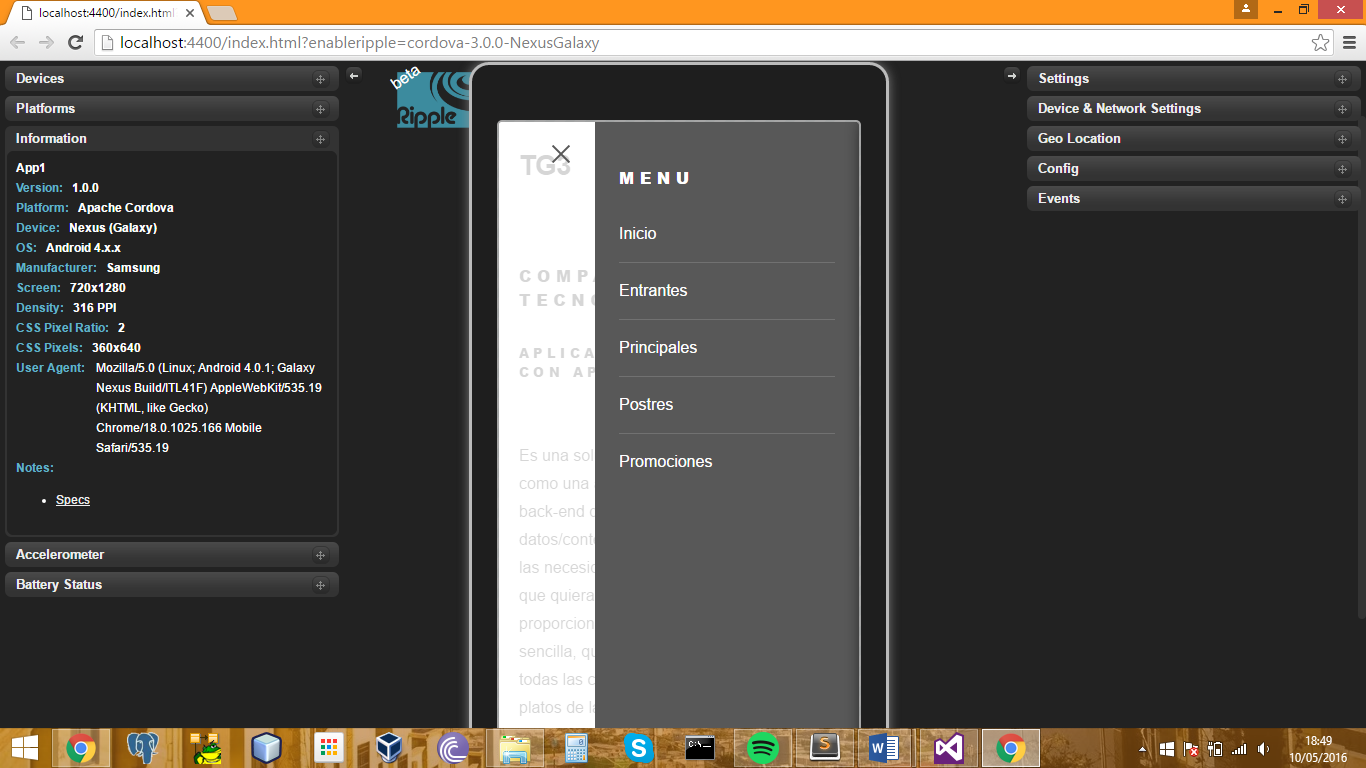
Una vez realizada la instalación, podemos crear proyectos de Apache Cordova.



Para poder emular el proyecto, o poder lanzarlo en un dispositivo hacemos uso de la solución que nos ofrece Visual Studio, pudiendo elegir que se realice una emulación entre varias plataformas y una serie de dispositivos, incluyendo la posibilidad de conectar nuestro propio dispositivo y lanzar la aplicación en el mismo.





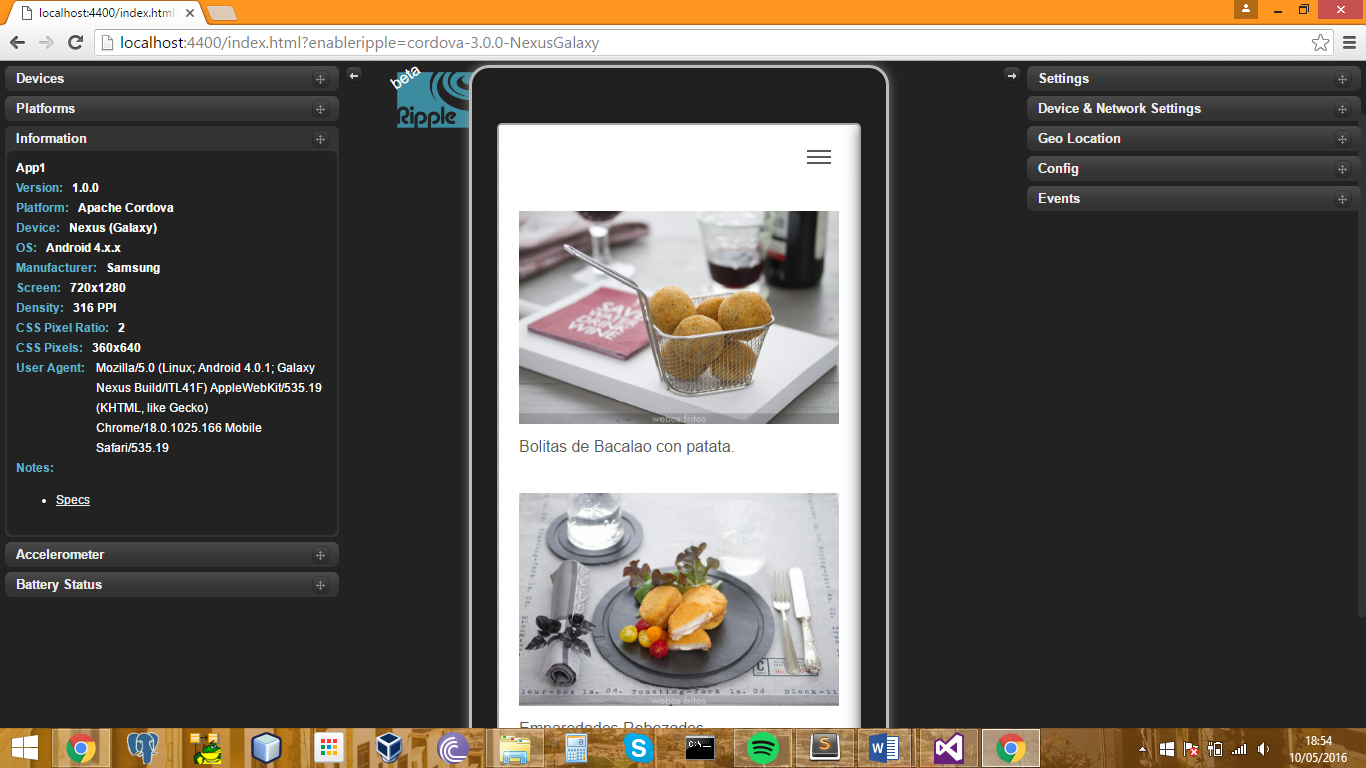


## Manual de usuario

La aplicación inicia con una primera ventana (Inicio), que contiene información principal junto con un icono que abre el menú principal

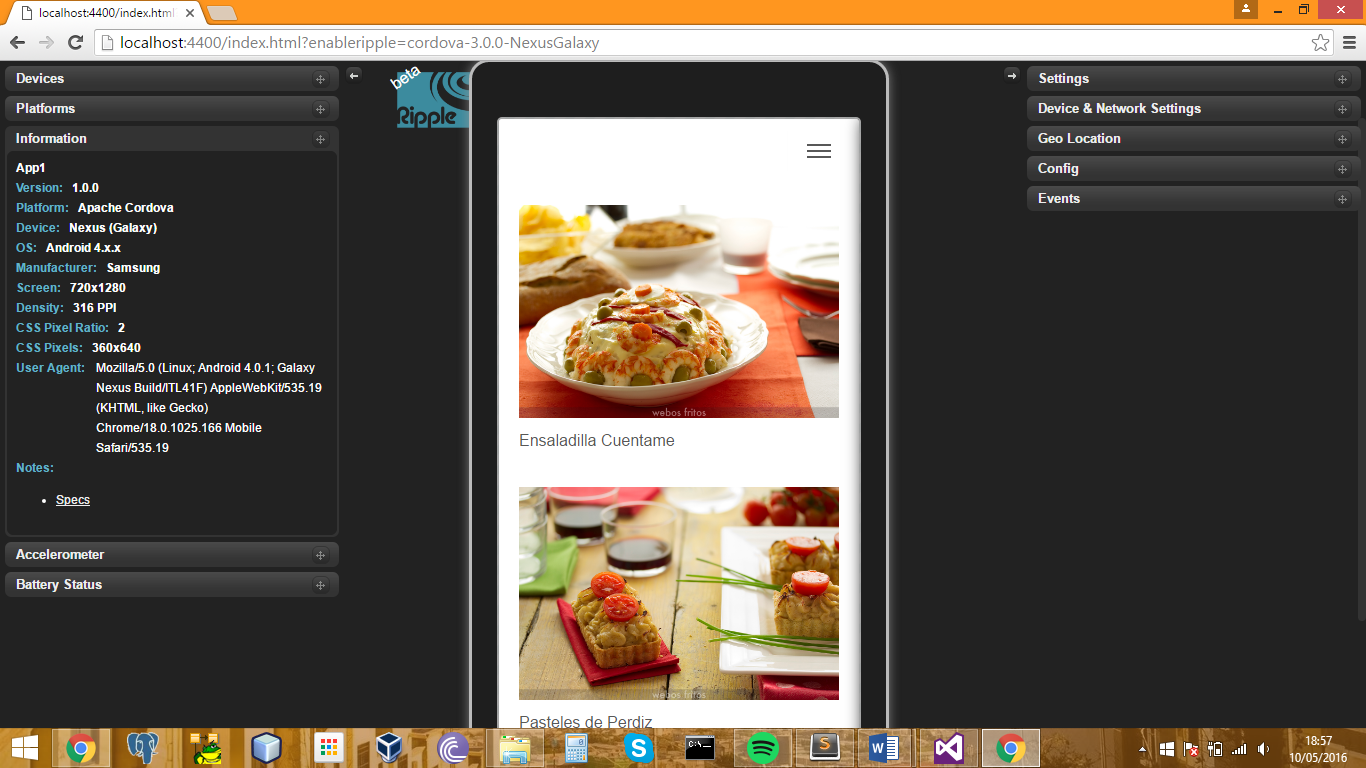
|  |  |
| --- | --- |
|  |  |

A través del menú principal podemos acceder a las demás pestañas de la aplicación:

* Entrantes:

En esta página encontraremos fotos de los entrantes con los que contamos, además de una pequeña descripción de cada uno a pie de página.

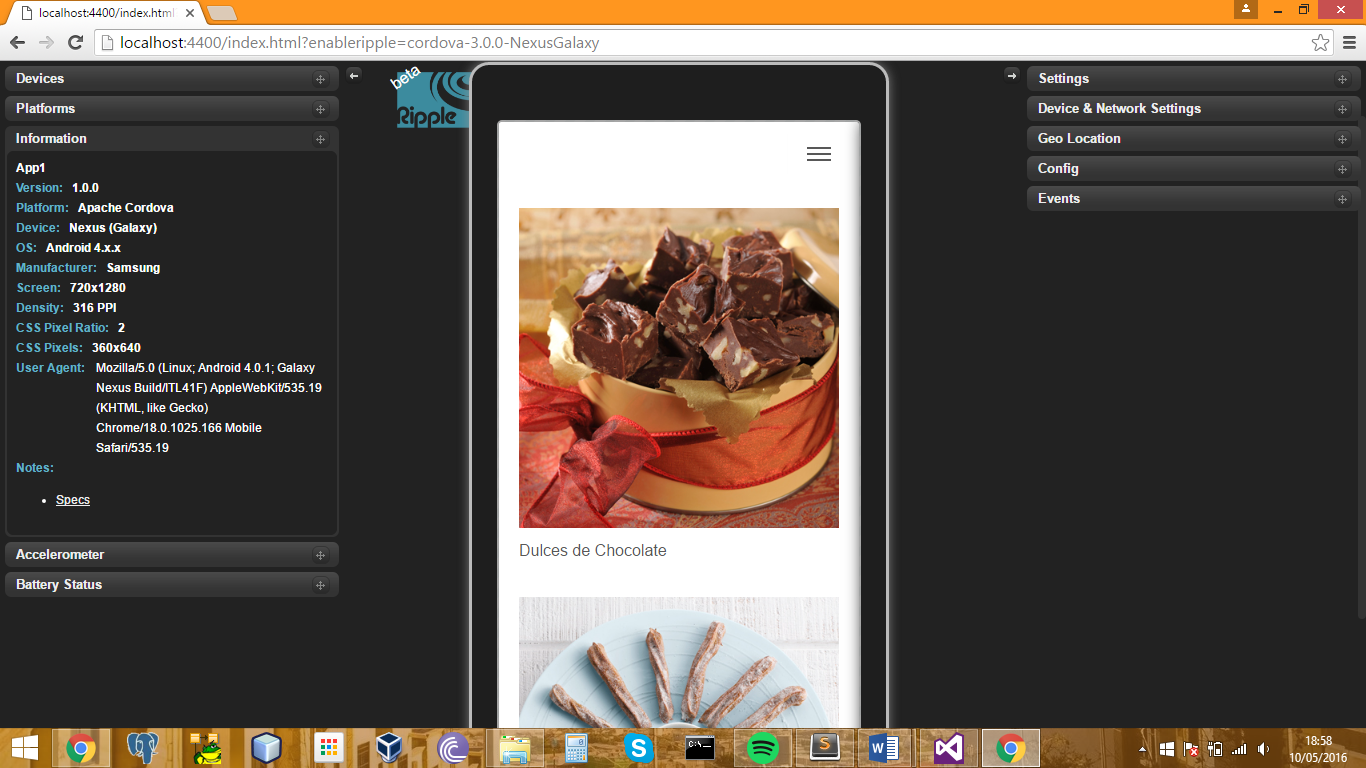
Como podemos apreciar, en la parte superior derecha seguimos teniendo el botón del menú principal. Este siempre se encontrará visible en esa posición.

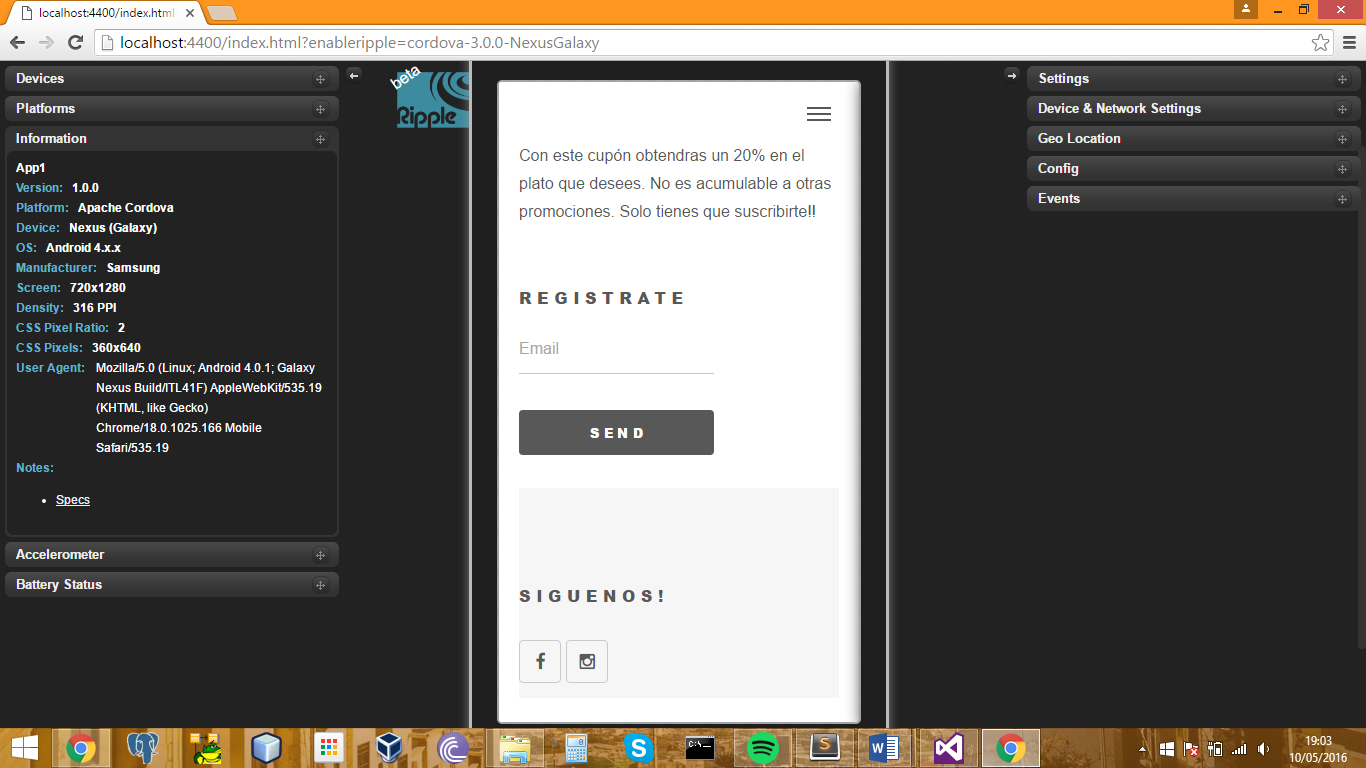
* Principales:

En esta página encontraremos fotos de los platos principales con los que contamos, además de una pequeña descripción de cada uno a pie de página.

Como podemos apreciar, en la parte superior derecha seguimos teniendo el botón del menú principal. Este siempre se encontrará visible en esa posición.

* Postres:

En esta página encontraremos fotos de los postres con los que contamos, además de una pequeña descripción de cada uno a pie de página.

* Promociones:

En esta página ofrecemos información sobre descuentos, además de incluir una sección para registrarse en la aplicación y así conseguir descuentos o información de forma individual.

En la parte inferior incluimos acceso a las redes sociales.

# Comparación de las dos implementaciones

## Evaluación de los criterios en la implementación usando Ionic

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Criterio 1: Horas empleadas en el desarrollo del sistema |  |
| Criterio 2: Curva de aprendizaje |  |
| Criterio 3: Herramientas externas |  |
| Criterio 4: Estructura del código |  |
| Criterio 5: Capacidad de adaptación |  |
| Criterio 6: Rendimiento |  |
| Criterio 7: Tiempo de arranque |  |
| Criterio 8: Recursos existentes en la red |  |

## Evaluación de los criterios en la implementación usando Apache Cordova

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Criterio 1: Horas empleadas en el desarrollo del sistema | 30 |
| Criterio 2: Curva de aprendizaje | 5 |
| Criterio 3: Herramientas externas | Si |
| Criterio 4: Estructura del código | 6 |
| Criterio 5: Capacidad de adaptación | 3 |
| Criterio 6: Rendimiento | 0.5 |
| Criterio 7: Tiempo de arranque | 23 |
| Criterio 8: Recursos existentes en la red | Abundante |

# Comparación de la implementación de las tecnologías

| **CRITERIOS** | **Ionic** | **Apache Cordova** | **COMENTARIOS** |
| --- | --- | --- | --- |
| Criterio 1: Horas empleadas en el desarrollo del sistema |  | 30 |  |
| Criterio 2: Curva de aprendizaje |  | 5 |  |
| Criterio 3: Herramientas externas |  | Si |  |
| Criterio 4: Estructura del código |  | 6 |  |
| Criterio 5: Capacidad de adaptación |  | 3 |  |
| Criterio 6: Rendimiento |  | 0.5 |  |
| Criterio 7: Tiempo de arranque |  | 23 |  |
| Criterio 8: Recursos existentes en la red |  | Abundante |  |

# Conclusiones

*A partir de la información incluida en el apartado 7 y de la experiencia al realizar el trabajo, el grupo debe estar en condiciones de manifestar su opinión sobre la implementación del sistema utilizando ambas tecnologías, y debe plasmarla en este apartado, indicando las ventajas e inconvenientes más relevantes de utilizar una u otra tecnología para implementar el sistema.*