

Guia de desarrollo híbrido mSSPA

Resumen

Contenido

- [Introducción](#)
- [Arquitectura](#)
 - [Clean Architecture](#)
 - [Patrón de Diseño – MVP](#)
 - [Patrón de Diseño - Repository](#)
 - [Arquitectura específica de mSSPA](#)
- [App Salud Andalucía](#)
- [Versión mínima del Sistema Operativo \(SO\)](#)
- [Entornos](#)
- [Versiones](#)
- [Frameworks](#)
- [Componentes y Tecnologías](#)
 - [Firebase](#)
- [Funcionalidades](#)
 - [Versiones](#)
 - [Menú lateral](#)
 - [Login](#)
 - [Logger](#)
 - [Evitar capturas de pantalla](#)
 - [Autenticación biométrica](#)
- [Seguridad](#)
 - [Comunicación segura](#)
 - [Dispositivos rooteados](#)
 - [Ofuscación de código](#)
- [Accesibilidad](#)
- [Internacionalización](#)
- [Tests](#)



Versión: v01r00

Fecha de publicación: 27 de enero de 2022

Fecha de entrada en vigor: 27 de enero de 2022

Introducción

La presente guía pretende ser un conjunto de buenas prácticas para el desarrollo de aplicaciones móviles híbridas que formarán parte del ecosistema del proyecto mSSPA.

Los componentes del proyecto mSSPA, a su vez, se basan en el resto de normativas definidas en la STIC, como por ejemplo, la de arquitectura y desarrollo para los componentes de backend, o la de interoperabilidad para la necesidad de consumo y definición de servicios que utiliza cada App.

A continuación se listan los enlaces a los distintos apartados ya existentes y disponibles públicamente en Confluence, para facilitar el acceso directo a los mismos:

Información sobre arquitectura y normativa de desarrollo: [Arquitectura](#)

Información sobre interoperabilidad: [Interoperabilidad](#)

Información sobre normativa de calidad: [Calidad](#)

Información sobre integración con las herramientas de gestión TIC: [Herramienta de gestión TIC - Servicios CGES](#)

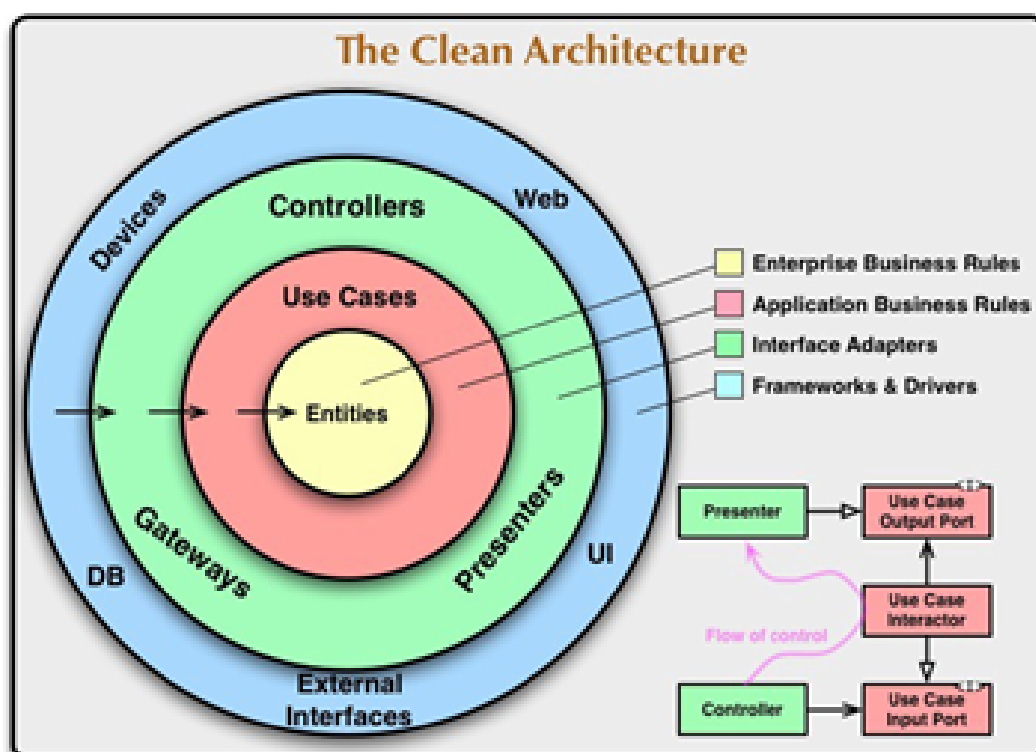
Arquitectura

Con objeto de garantizar la robustez, calidad y mantenibilidad de la aplicación, así como una buena estructura del código de la aplicación, es muy importante definir una buena arquitectura. En este apartado se describe la arquitectura adoptada.

Clean Architecture

La arquitectura a usar por la apps será Clean Architecture, una arquitectura de uso extendido en el desarrollo nativo de aplicaciones móviles. Entre las ventajas de usar **Clean Architecture** podemos citar:

- Separación del código en capas independientes con responsabilidades bien definidas para cada una de las capas.
- Mayor nivel de abstracción.
- Código desacoplado.
- Facilita el uso de tests.



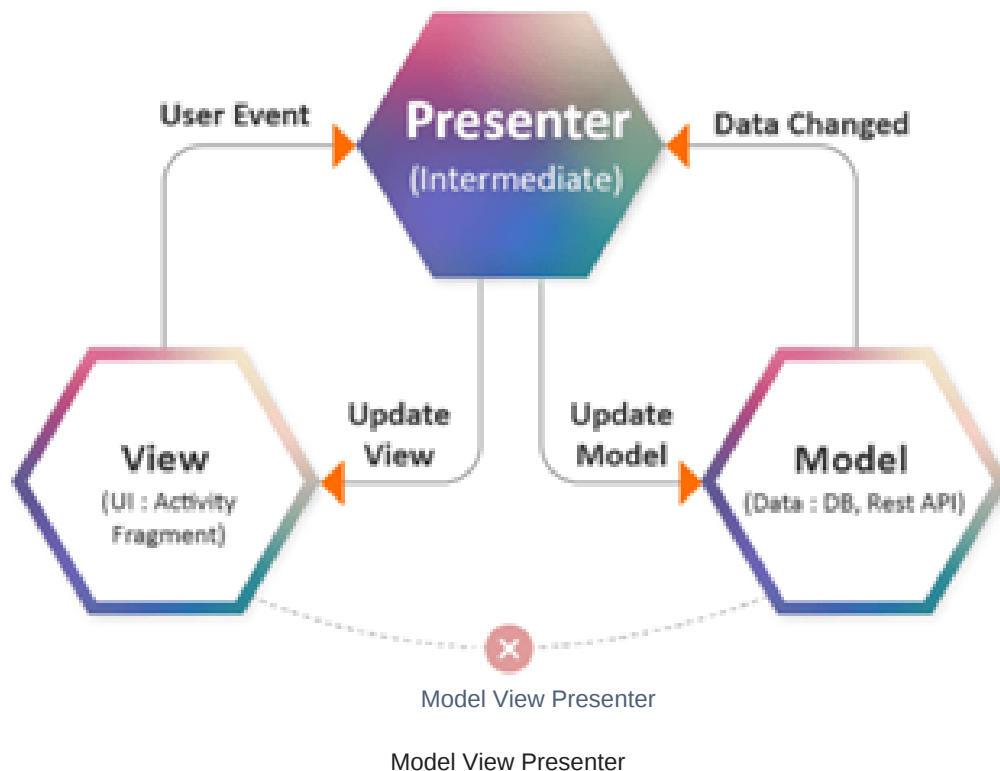
Clean Architecture

Clean Architecture

Patrón de Diseño – MVP

Uso del patrón de diseño **MVP** (Model View Presenter). Patrón de diseño muy utilizado en el sector del desarrollo móvil nativo. Dicho patrón separa la capa de la vista o presentación con la capa de negocio, haciendo de esta manera el código más robusto, mantenible y testeable.

Android MVP (Model View Presenter)

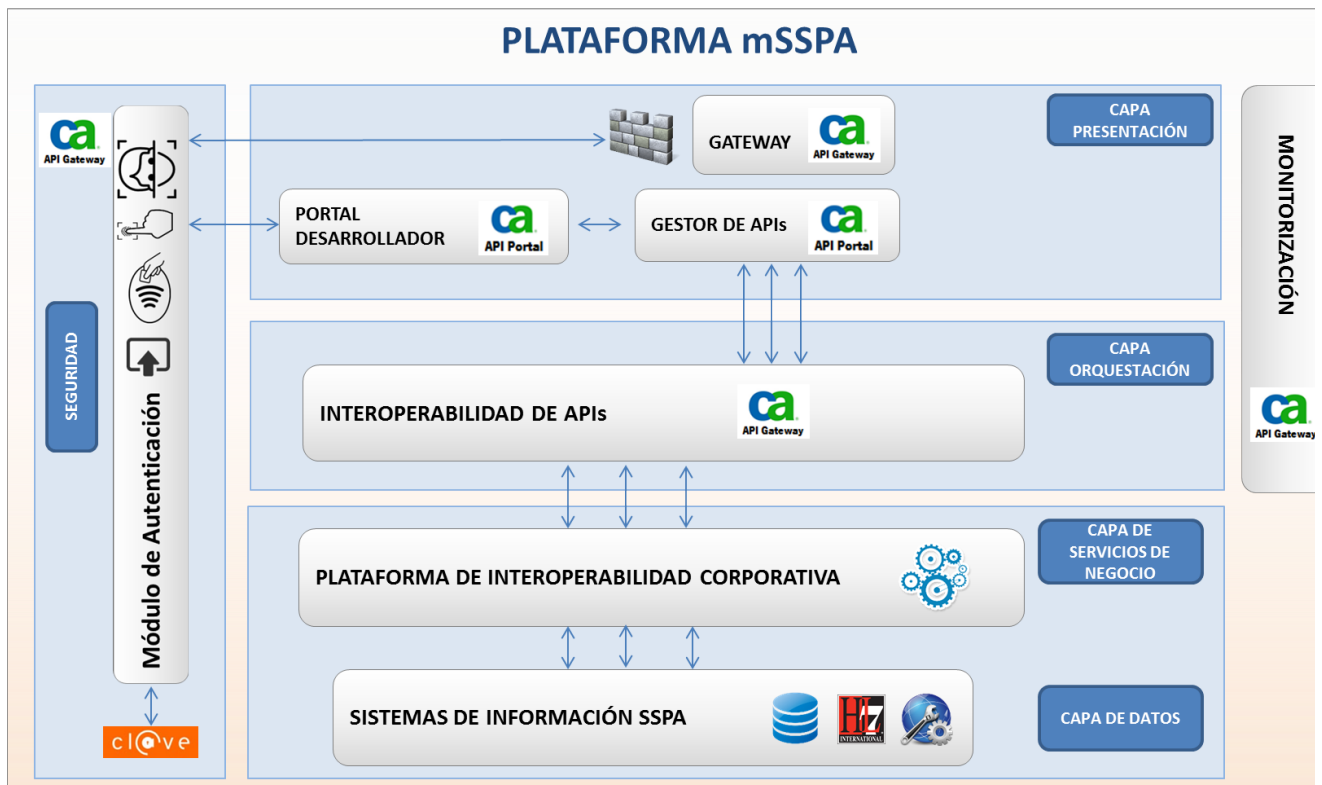


Patrón de Diseño - Repository

Una de las ventajas de tener el código de la app separado en capas es la independencia de cada una de las capas en cuanto a responsabilidades. En este sentido, en la capa de acceso a datos haremos uso del patrón de diseño Repository. Este patrón nos permite separar la fuente de datos del resto de partes de la aplicación permitiendo en caso necesario interactuar con una o varias fuentes de datos distintas e incluso cambiar la fuente de datos utilizada por la aplicación.

Arquitectura específica de mSSPA

mSSPA plantea la arquitectura que se describe a continuación:



App Salud Andalucía

App referente y aglutinador de todas las apps del SSPA que le da acceso a toda información de dicho organismo, incluye una sección específica sobre el coronavirus Covid-19, tiene un sistema de consulta y seguimiento, un sistema de lanzamiento de campañas de seguimiento, y además cuenta con un agente conversacional.

Como funcionalidades principales:

- Listado de noticias
- Listado de Aplicaciones desarrolladas dentro del ecosistema del SSPA
- Funcionalidades relacionadas con Covid-19 (certificado Covid-19, autotriaje Covid-19, etc.)
- Recepción de Notificaciones push
- Acceso a las funcionalidades de ClicSalud+
- Red de centros SSPA
- Seguimiento de pacientes crónicos

Versión mínima del Sistema Operativo (SO)

Se propone cómo mínima versión soportada la versión 5.0 Lollipop (API Level 21) en caso del SO Android y iOS 12 para el caso de iOS. Según los últimos datos de adopción de los SOs, aproximadamente el 90% de los dispositivos existentes tienen instalada al menos dicha versión o una superior.

Entornos

Es una buena práctica en el diseño de cualquier proyecto software separar en entornos de desarrollo distintos las distintas fases del ciclo de vida del software, evitando así datos incongruentes o erróneos en entornos productivos. Se contará con dos entornos de desarrollo distintos, uno para la fase de desarrollo al que llamaremos entorno PRE y otro para la fase de producción al que llamaremos PRO, y se hará uso de configuraciones distintas para los entornos mencionados de PRE y PRO.

Versiones

Al igual que en el caso de los entornos de desarrollo, también se diferenciarán las versiones de compilación. De esta forma como mínimo existirá una versión de compilación de “debug” y otra “release”.

Será la versión “release” la que habrá que seleccionar para cuando se quiera crear una versión funcional de la aplicación que esté lista para la subida a la tienda de aplicaciones.

Frameworks

Se hará uso de los frameworks Ionic con Angular y Cordova. A ser posible, estará actualizado a las última versiones estables de dichos frameworks.

Ionic es un framework de código libre con componentes de UI para contruis aplicaciones móviles basadas en tecnologías webs, tales como HTML, CSS y JavaScript. Junto con el framework de Ionic se usará el framework de Angular, ya incluido en el propio framework de Ionic.

Además de Ionic, se hará uso del framework Cordova. Cordova es un framework que facilita el uso de las capacidades nativas de cada dispositivo móvil, independientemente de la plataforma. Así expone APIs de acceso a la cámara del dispositivo, al gps, etc.

Componentes y Tecnologías

Firebase

Firebase es un conjunto de herramientas y servicios proporcionadas por Google que facilitan ciertas tareas de análisis, configuración, comunicación entre otras para con las distintas aplicaciones móviles que lo integren. De todos los servicios que ofrecen, los mínimos que deben integrarse son los siguientes:

Analytics

Es importante conocer el uso que un usuario hace de cualquier aplicación una vez la app ha sido liberada. Es interesante medir tiempos de sesión, uso de ciertas funcionalidades, etc. En este sentido, mientras más datos se conozcan sobre el uso de la aplicación más argumentos de decisión se tendrán para actuar en consecuencia.

Analytics permite definir tanto eventos como propiedades de los usuarios que formarán parte de las estadísticas de uso de la aplicación.

Crashlytics

Al igual que el apartado anterior, es de vital importancia conocer los errores de la aplicación en tiempo real. Crashlytics proporciona información sobre los errores que se han producido en la ejecución de la aplicación por parte de los usuarios, aportando información técnica muy valiosa que permite disponer de una traza de en dónde y qué ha provocado el error, permitiendo así solventar esos posibles errores en el menor tiempo posible.

Funcionalidades

En este apartado se describen las funcionalidades mínimas comunes para todas las aplicaciones que formarán parte del ecosistema del proyecto mSSPA.

Versiones

Tal y como se ha detallado en apartados anteriores, se contará con dos versiones distintas, la versión “debug” y la versión “reléase”, además de contar también con dos entornos de desarrollo distintos, PRE y PRO.

Con el objetivo de poder diferenciar claramente el contexto en el que se está ejecutando la app se debe diferenciar el nombre de la aplicación, así como su nombre de paquete, incluyendo el sufijo “DEV”, permitiendo en este sentido incluso poder instalar ambas versiones (“debug” y “reléase”) de la app en el mismo dispositivo.

Menú lateral

Las aplicaciones incluirán el elemento Drawer Layout para proporcionar un menú lateral con accesos directos a distintas secciones de la aplicación.

Además de accesos directos a secciones, el menú lateral deberá de informar de la versión de la aplicación que se está ejecutando, además del entorno de desarrollo. Para ello, en caso de que la aplicación se esté ejecutando en el entorno de desarrollo de PRE se incluirá un texto indicativo a la versión de la aplicación.

Login

Desde las aplicaciones podremos autenticar al usuario contra los sistemas de información del SAS de forma que tras la autenticación obtengamos un token de acceso para sucesivas llamadas a dichos sistemas.

Para este propósito las apps se integrará con el nuevo sistema de login basado en WebView a través del producto de mSSPA denominado "**Web de Opciones de Autenticación**", que actualmente contiene como opciones de login Certificado Digital, Cl@ve, WebMovil SSPA (login con QR) y DMSAS (login profesional).

Esta integración se llevará a cabo haciendo uso de la API y servicios expuestos a través de CA para la posterior obtención del token de acceso para sucesivas peticiones para la obtención de datos.

Deeplinking

Las aplicaciones pertenecientes al proyecto mSSPA deben tener la capacidad de abrirse desde enlaces externos desde otras aplicaciones tales como el gestor de correo o otra aplicación del proyecto mSSPA.

Se debe establecer, de forma clara, las urls que las aplicaciones sabrán interpretar como links externos para poder abrir la aplicación o la sección correspondiente en cada momento.

Logger

Con el objetivo de poder conocer los eventos y las posibles anomalías que pudiesen suceder, se propondrá un sistema de Logger para identificar claramente qué y dónde se ha producido el evento o anomalía.

Así mismo se deberá evitar escribir mensajes en los logs en las versiones de las aplicaciones que se publiquen en el store ya que estos datos sólo son útiles durante el desarrollo de la aplicación. Además, también se evitará la escritura de datos confidenciales del usuario.

Ocultar información en segundo plano

Debido a la confidencialidad de la información con la que las aplicaciones del proyecto mSSPA trabajarán, es de debido cumplimiento que las aplicaciones oculten la información mostrada cuando dichas aplicaciones pasen a segundo plano.

Evitar capturas de pantalla

Al igual que el apartado anterior, las aplicaciones del proyecto mSSPA no permitirán que se puedan realizar capturas de pantalla para evitar así filtrar información confidencial del usuario.

Autenticación biométrica

Cada vez son más los dispositivos con capacidades biométricas integradas, como lector de huella o reconocimiento facial.

Las aplicaciones dispondrán de autenticación biométrica siempre y cuando el dispositivo lo permita. Este método de autenticación sentará las bases para que futuras funcionalidades puedan hacer uso de la autenticación biométrica en caso de ser requerida.

Seguridad

La seguridad es un aspecto clave debido a la sensibilidad de los datos a tratar en las aplicaciones del proyecto mSSPA. En este sentido se establecen una serie de medidas mínimas de seguridad tales como:

Comunicación segura

Cualquier comunicación con el servidor vía API deberá ser una comunicación segura bajo protocolo HTTPS.

Dispositivos rooteados

Las apps harán las comprobaciones pertinentes para detectar si el dispositivo ha sido “rooteado” o no. Un dispositivo “rooteado” es un dispositivo en el que el usuario tiene control total sobre los privilegios de todas las aplicaciones y procesos que se ejecutan en el mismo, por lo que podría tener acceso a los datos sensibles del usuario.

Ofuscación de código

Se establecerán medidas para la correcta ofuscación del código de la app siguiendo las directrices que marque el SO.

Accesibilidad

Cualquier app sigue unas pautas de implementación de capacidades de accesibilidad para hacerla usable a personas con algún tipo de limitación, ya sea auditiva o visual. Para ello, deberá seguir la normativa que provee la Oficina de Calidad.

Internacionalización

Aunque el ámbito de las aplicaciones del proyecto mSSPA está claramente definido a la Comunidad Autónoma de Andalucía, además del castellano, las apps también soportarán el inglés como idioma principal de la aplicación, facilitando con esto su integración con proyectos internacionales.

Tests

La estabilidad y calidad del código es una pieza fundamental dentro de cualquier proyecto software que se desarrolle.

Para garantizar la robustez y calidad del código, se establecerán test (unitarios, funcionales o de integración) en las distintas capas de la arquitectura de la aplicación que sean capaces de garantizar un buen porcentaje de cobertura de código.