* Seguir los estándares de nomenclatura plasmados en el presente documento.
* Cada rol debe cumplir las actividades designadas en el documento.
* Cada miembro debe trabajar únicamente y exclusivamente sobre su branch.
* La filtración de información a agentes externos se encuentra estrictamente prohibida.
* Correcta comunicación entre los miembros encargados de la gestión de la configuración.
* Toda dependencia en los proyectos deberá ser estrictamente documentada.
* Se encuentra estrictamente prohibido el versionamiento en el nombre del ítem. Ej.: No se permite el nombre “SPC\_AC\_v1.1.docx”
* Para los ítems de gestión de la configuración:
  + Los ítems deberán llevar el siguiente estándar de nombre:

<INICIALES DEL DOCUMENTO>.\*

Ej.: PGC.docx (Plan de gestión de la configuración)

* En caso dos ítems de gestión de la configuración llegasen a tener la misma nomenclatura, el ítem más reciente tendrá la nomenclatura:
  + <INICIALES DEL DOCUMENTO>\_<TAG>.\*

Donde <TAG> es un identificador para dar soporte a la identificación del documento más reciente.

* Para los ítems de gestión de proyectos:
  + Los ítems deberán llevar el siguiente estándar de nombre:

<INICIALES DEL PROYECTO>\_<INICIALES DEL DOCUMENTO>.\*

Ej.: SPC\_AC.docx

* Los nombres de los branches deberán ser hecho según la inicial del nombre del responsable y del primer apellido completo. Ej.: Eduardo Angulo -> branch “EAngulo”
* Para los ítems de desarrollo de proyectos Frontend:
  + Todo folder o ítem que cuente con más de 2 palabras, deberá ser nombrado en CamelCase. Ej.: Dashboard home-> DashboardHome
  + Los componentes:
    - Los ítems que expongan los componentes deberán ser nombrados de la siguiente manera:

<NOMBRE DEL COMPONENTE O FEATURE>component.js

* + Las vistas o templates:
    - Deberán ir en el folder de su respectivo componente o feature (característica de la aplicación). En este caso se utilizará el Views como carpeta de las vistas.
    - Dentro de la carpeta Views tendremos las carpetas Home y Shared. Shared nos permitirá tener las vistas compartidas como el \_Layout y la vista de carga de datos en otra vista así como también la vista de Errores.
    - Deberán ser nombradas de la siguiente manera:

<NOMBRE DEL COMPONENT O FEATURE>Index.cshtml

* + Los controladores
    - Deberán ir en el folder de su respectivo componente o feature (característica de la aplicación).
    - Deberán ser nombradas de la siguiente manera:

<NOMBRE DE COMPONENT O FEATURE>HomeController.cshtml

* + Los services(servicios)
    - Deberán ir en el folder src/app/core del proyecto.
    - Dependiendo de la utilidad del servicio, este deberá ir en la siguiente carpetas:
      * Utils.- servicios que utilizan componentes.
      * Helpers.- servicios para evitar la repetición de código o snippets.
      * Services.- servicios con lógica de procesos.
    - Los ítems que expongan los servicios, deberán ser nombrados de la siguiente manera:

<NOMBRE DEL SERVICIO>.service.js

* + Los resources(recursos de datos)
    - Deberán ir en el folder src/app/core/resources del proyecto.
    - Dependiendo de su fuente, deberán ir en las carpetas:
      * Local.- Extracción de datos almacenados en el navegador.
      * Test.- Para mock.
      * Remote.- Extracción de datos de fuentes remotas(APIs).
  + Los estilos de la aplicación
    - Deberán ir en src/assets.
    - Se deberá seguir el patrón de diseño 7-1. Ref: <https://sass-guidelin.es/es/#el-patron-7-1>
  + Imágenes y documentos
    - Deberán ir en las carpetas src/images y src/docs respectivamente.
* Para los items de proyectos BackEnd:

Todo ítem con más de dos palabras deberá ser nombrado en Camel case. Ejm: controlador de usuario -> UserController

* + Será usado como estructura del proyecto el generado por el Framework Laravel:
  + El directorio raíz
    - El directorio app
    - El directorio bootstrap
    - El directorio config
    - El directorio database
    - El directorio public
    - El directorio resources
    - El directorio routes
    - El directorio storage
    - El directorio tests
    - El directorio vendor
  + El directorio app
    - El directorio Console
    - El directorio Events
    - El directorio Exceptions
    - El directorio Http
    - El directorio Jobs
    - El directorio Listeners
    - El directorio Mail
    - El directorio Notifications
    - El directorio Policies
    - El directorio Providers
    - El directorio Rules
* Para los ítems de proyectos móvil:
  + Todo folder el cual contenga más de 2 palabras como por ejemplo: android Test, será nombrado del siguiente modo: androidTest. En resumen del siguiente modo: <nombreDeLaCarpeta>
    - Solo en el caso de tratarse de las carpetas de los componentes gráficos tendrán esta estructura:

En resumen del siguiente modo: <drawable-hdpi>

* + Los componentes serán nombrados del siguiente modo:

Las clases hechas en JAVA las son expresadas del siguiente modo:

<NombreDeLaClase>.java

* + Los layouts y componentes gráficos o vistas se organizarán del siguiente modo:

Los archivos xml están con la siguiente nomenclatura: <activity\_main>.xml

Los archivos como fotos o imágenes tienen la siguiente nomenclatura: <ic\_add\_without\_circle>.png

* + Todo archivo que contenga configuración del mismo proyecto Android como el graddle o los services de Google API terndrán la siguiente nomenclatura:

<nombre-archivo>.<extension>

Ejm:

google-services.json

proguard-rules.pro

* + - * + Todas las imágenes se encuentran distribuidas en la carpeta <nombre- app>/<app>/<src>/<main>/<res>, y dependiendo su dimensión se pueden ubicar en drawable, drawable-nodpi, etc.