1. Diseñe una arquitectura CSS:

Distribución:

* La distribución iría en dos estructuras la primera sería un estilo de estado (clases globales y fijas) donde crearía una serie de carpetas que contengan estilos generales como: alineación y distribución, colores, tamaños de botones, diseño de tablas, diálogos, calendarios y demás propiedades globales, las cuales serían importadas a la hoja de estilos principal del proyecto. La segunda estructura iría en función propia de cada módulo o componente, en caso de que sobre un módulo especifico apliquen unas clases propias estas serían distribuidas por carpetas e importadas sobre los componentes a los que les aplique esas clases. Por último, toda interacción del [ngClass] sería sobre las clases propias del componente, así como el diseño resposive.

Procesador:

* SASS, debido a que ofrece la opción de manejar variables globales sobre las clases generales y permite además crear iteraciones y condiciones para personalizar el comportamiento del diseño en función de la interacción con el usuario y por último ofrece la facilidad de definir el comportamiento de una clase en un solo llamado, independiente de que su comportamiento cambie en función de la resolución (los media query se definen dentro de la clase).

Hack important:

* Al tener una serie de niveles de clases sobre los diferentes elementos del DOM el hack important sirve para primar la definición de una propiedad sobre otra, por ejemplo, si un elemento de un componente tiene definido un tamaño de texto de 14px de acuerdo a lo definido en la clase global y se requiere que el tamaño sea 16px es posible definir otra clase que cambie esa propiedad y por medio de hack important este comportamiento prima sobre el anterior.

Themes:

* Sobre la hoja de estilos global es posible importar un tema personalizado o uno ya definido sobre este irían todos los comportamientos globales, es decir sobre la definición del tema iría la primera estructura definida en el punto uno.

Como procesador usaría SASS debido a que ofrece la opción de definir variables globales, además de permitir crear iteraciones y condicionales dentro del código que permite programar personalizaciones de manera más fácil y ágil en comparación a programarlo sobre el javascript puro. Lo primero sería definir un tema general donde se alojarían los diseños globales como, tamaños, colores, botones, tablas, diálogos, clases flex-box y demás comportamientos fijos, luego por cada modulo dependiendo la necesidad se crearía una serie de clases que controlarían el comportamiento de los componentes y sobre estos últimos trabajaría únicamente con las clases que tengan definido comportamientos dentro de la directiva [ngClass] o alguna iteración dentro de un media query. Esta estructura permite ahorrar código, además de hacer más fácil la identificación y el cambio de alguna clase general, la interacción del usuario es fácil relacionarlo dentro de la clase ya que esta es alojada solo en el componente de presentación. La principal técnica usada es la definida en la guía SMACSS.