

embarcadero®

Delphi Academy

Consejos prácticos, trucos y técnicas



Primeros pasos con **Ext JS**

Fernando Rizzato
Lead Software Consultant, Latin America

AGENDA

- Ext JS Toolkits
- MVC / MVVM
- Layouts y Containers
- Data Packages
- Demos

EXT JS TOOLKITS

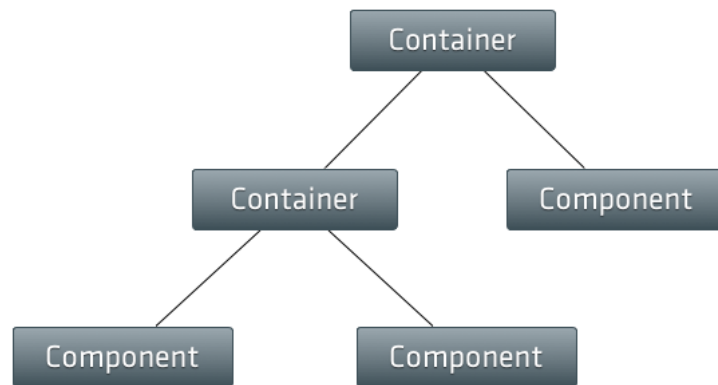
- Classic
 - para los navegadores desktop/tablet
- Modern
 - para los navegadores modernos/móviles
- Universal
 - soporta ambos Classic y Modern

MVC / MVVM

- La principal diferencia entre **MVC** y **MVVM** es que el **MVVM** presenta una abstracción de una vista llamada ViewModel. ViewModel coordina los cambios entre los datos de un modelo y la presentación de ese dato usando una técnica llamada "data binding"
- **(M)** Model - Estos son los datos para su aplicación. Los modelos saben cómo persistir en el paquete de datos y pueden vincularse a otros modelos a través de asociaciones.
- **(V)** View - La View es cualquier tipo de componente que se representa visualmente. Por ejemplo, los *grids*, los *trees* y los *panels* se consideran Views
- **(C)** Controller - Los controladores se utilizan como una ubicación para mantener la lógica de la visión que hace que la aplicación funcione. Esto puede implicar la construcción de vistas, enrutamiento, la creación de instancias de modelos y cualquier otro tipo de lógica de aplicación
- **(VM)** ViewModel - ViewModel es una clase que administra datos específicos para la vista. Permite que los componentes interesados se enlacen a él y se actualicen cada vez que se cambian los datos

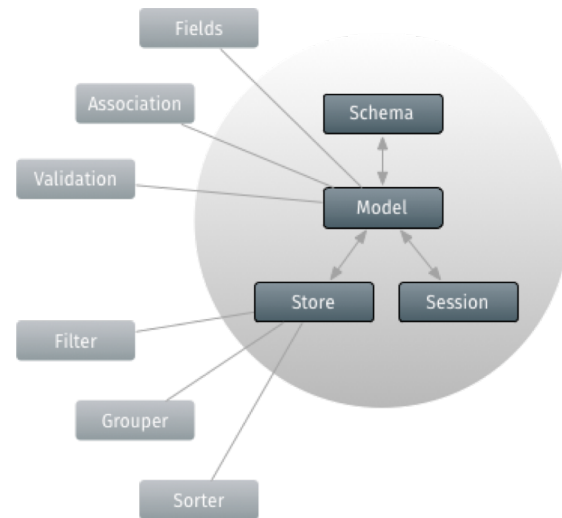
LAYOUTS Y CONTAINERS

- El sistema de diseño es una de las partes más poderosas del Ext JS. Se ocupa del dimensionamiento y posicionamiento de todos los componentes en su aplicación
- Una UI de una aplicación Ext JS se compone de componentes. Un contenedor es un tipo especial de componente que puede contener otros componentes. Una aplicación Ext JS típica se compone de varias capas de componentes anidados



DATA PACKAGES

- El paquete de datos es el que carga y guarda todos los datos en su aplicación
- La pieza central del paquete de datos es **Ext.data.Model**. Un **Model** representa una entidad en una aplicación. Por ejemplo, una aplicación de comercio electrónico puede tener modelos para usuarios, productos y pedidos
- **Proxies** son usados por los **Models** y **Stores** para manejar la carga y el rescate de datos del modelo. Hay dos tipos de *proxies*: Cliente y Servidor
- Los **Models** son normalmente usados por un **Store**, que es básicamente una colección de registros (instancias de una clase derivada de **Model**)



DEMOS

RECURSOS ADICIONALES

Documentación

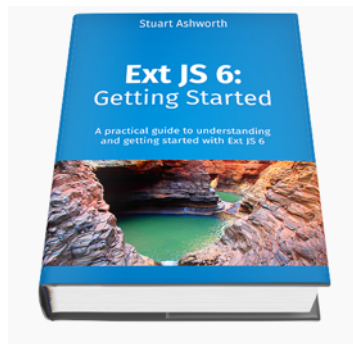
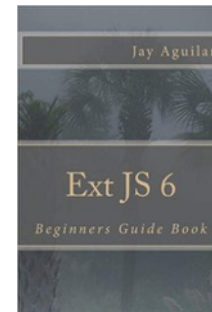
- https://docs.sencha.com/extjs/6.7.0-CE/guides/other_resources/oop_concepts.html
- https://docs.sencha.com/extjs/6.7.0-CE/guides/getting_started/open_tooling.html
- https://docs.sencha.com/extjs/6.7.0-CE/guides/core_concepts/classes.html
- https://docs.sencha.com/extjs/6.7.0-CE/guides/core_concepts/layouts.html
- https://docs.sencha.com/extjs/6.7.0-CE/guides/core_concepts/data_package.html
- <https://examples.sencha.com/extjs/6.7.0/>

Blogs

- <https://linuxize.com/post/how-to-install-node-js-on-ubuntu-18.04>
- <https://www.swarmonline.com/20-tips-for-getting-started-with-ext-js/>
- <https://www.extjs-tutorial.com/>
- <https://www.tutorialspoint.com/extjs/index.htm>
- https://www.youtube.com/playlist?list=PLIUDMEtO0HdS4_YbmTAU7GyygSBCPpUu

EXT JS BOOKS

- <https://www.extjs6gettingstarted.com/>
- <https://www.oreilly.com/library/view/ext-js-6/9781783550494/>
- <https://www.amazon.com/Sencha-Ext-JS-Bootcamp-Book/dp/1539406660>
- <https://www.amazon.com/Mastering-Ext-JS-Loiane-Groner/dp/1782164006>
- <https://www.amazon.com/Ext-JS-Beginners-Guide-Book/dp/1539380858/httpwwwwuto0a-20>
- Packt Publishing: <http://bit.do/eT8T9>



GRACIAS!

Preguntas?

Me puedes encontrar en:

@FernandoRizzato

fernando.rizzato@embarcadero.com

Síguenos en

fb.com/EMBTLatAm