# 3. Conceptos teóricos

En este capítulo se van a tratar de forma superficial todos los conceptos teóricos que actuarán a modo de base para comprender el proyecto.

## 3.1 Plotly

Plotly Python es una librería que genera gráficos interactivos en HTML. Es interesante mencionar brevemente dos de sus módulos:

* Plotly Express (usualmente importada como px): API de alto nivel para visualizar datos.
* Plotly Graph Objects (usualmente importada como go): API de bajo nivel para figuras, trazos y diseño.

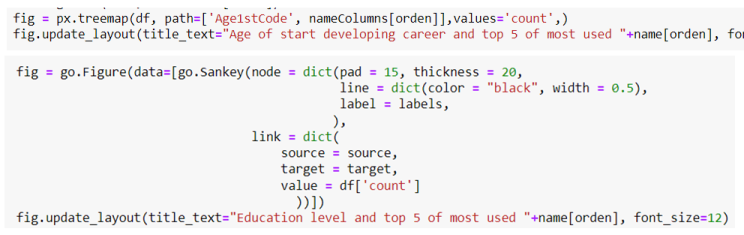


Ilustración 1 Ploty Express vs Plotly Graph Objects

Plotly Express tiene una sintaxis significativamente más sencilla, pero sus gráficos son menos llamativos visualmente.

## 3.2 Dash.

Dash, tal y como es definido en su documentación[[1]](#endnote-1), es un marco para crear rápidamente aplicaciones de datos en Python. Está diseñado para ser usado con Plotly Python. Tanto es así que a veces es denominado Plotly Dash. Es ideal para crear e implementar aplicaciones de datos con interfaces de usuario personalizadas. Es particularmente adecuado para cualquiera que trabaje con datos y no sea muy experimentado en HTML.

A continuación, se describirán los componentes más importantes que se han usado para construir la aplicación:

### 3.2.1 Dash HTML y conceptos básicos en HTML.

Aunque no se ha programado directamente en web usando exclusivamente HTML o CSS, sí se han necesitado estudiar algunos conceptos básicos. Dado que la programación web no ha sido uno de los apartados más importantes en el proyecto, tan solo se va a mencionar muy brevemente las características comunes de HTML con el módulo Dash HTML Components – módulo en el que se ha apoyado para construir el layout de la página.

La mayoría de elementos HTML usados en este dashboard constan de tres partes muy diferenciadas: una etiqueta de apertura, un contenido y una etiqueta de cierre. En Dash se consigue el mismo efecto mediante el uso de corchetes y paréntesis.

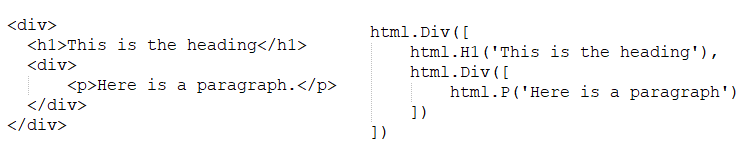


Ilustración 2 Comparación HTML y Dash: conceptos básicos

La estructura al añadir propiedades es similar al usar componentes HTML y Dash, pero hay algunas diferencias clave[[2]](#endnote-2):

* La propiedad style es un diccionario.
* class es renombrada como className.
* Se pueden omitir los píxeles en las propiedad de style.

Dash tiene un componente capaz de imitar a cada uno de HTML. Por ejemplo, el componente Graph ha sido usado en la aplicación para renderizar todos los gráficos generados por Plotly.

Se puede ampliar esta información y ver una lista que profundice en detalle sobre cada uno de ellos en la documentación de Dash.

### 3.2.1 Dash Core Components.

Componentes que permiten crear interacción con la página. La documentación[[3]](#endnote-3) puede nombrar varios ejemplos, pero en la aplicación se ha usado especialmente el componente dropdown.

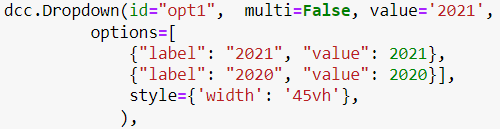


Ilustración 3 Dash Core Components: elemento dropdown

Para crear esa interacción, se usan las funciones callback. Se profundizará en ellas en el [apartado 3.2.1](#_3.2.1_Funciones_callback.) de este documento.

### 3.2.3. Dash Bootstrap Components.

Dash Bootstrap es una biblioteca de componentes de Bootstrap que facilita la creación de aplicaciones con diseños responsive.[[4]](#endnote-4) Estos componentes se han usado especialmente para diseñar el layout del dashboard, pero no se ha profundizado demasiado en ellos.

## 3.2. Dash y el modelo - vista – controlador.

Dash, tal y como lo define su propia documentación, es un framework escrito sobre Plotly y React y es ideal para crear e implementar aplicaciones con interfaces personalizadas.

Todas las aplicaciones creadas con Dash, incluyendo la creada en estre proyecto, siguen una estructura de tres partes consecutivas: manipulación de datos, creación del layout e interacción entre los componentes. Esta particular estructura puede asociarse fácilmente con el modelo – vista – controlador[[5]](#endnote-5) de la siguiente forma:

Modelo, acorde con el modelo – vista – controlador (MVC), únicamente representa los datos. Si se tiene en cuenta la estructura de tres partes mencionada anteriormente se puede llegar a la conclusión que su equivalente en Dash sería la **manipulación de datos**.



Ilustración 4 Ejemplo manipulación datos/modelo.

**Vista** permite que el usuario pueda visualizar los datos del modelo. Se incluirá únicamente el aspecto visual de todos los botones, gráficos y demás elementos gráficos. En Dash corresponderá con la **creación del layout**.

En el proyecto se han usado dos componentes de la librería Dash llamados Dash Core Components y Dash Bootstrap Components. Estos se verán con más profundidad en el apartado siguiente.

**Controlador** crea todas las interacciones entre los elementos del layout. En Dash y en la aplicación, se asociaría con la **interacción de los componentes**. Es decir: con la creación de las funciones callback.

### 3.2.1 Funciones callback.

Estas funciones son llamadas de forma automática cada vez que se cambia la propiedad de un elemento de entrada para actualizar alguna propiedad en un componente de salida.[[6]](#endnote-6)

En el apartado anterior se mencionó como, en el layout, el componente Graph tomaba como input una figura generada con plotly. Las funciones callback funcionan de forma similar.

Todas las funciones callback necesitan un output y un input. Estos a su vez necesitan un component\_id y un component\_property. Cada uno de estos componentes en el callback referenciará un componente en el layout.

Se explicará este concepto sobre dos fragmentos de código de la aplicación, el primero perteneciendo al layout y el segundo a su propia función callback:

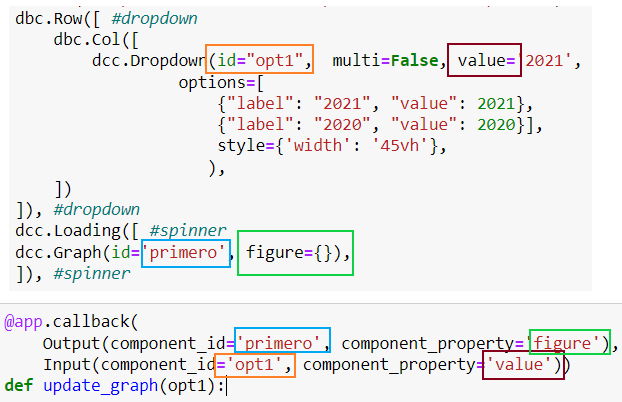


Ilustración 5 Funcionamiento de callback en fragmento de código

El output de la función callback que se ve en la imagen es el gráfico generado en Plotly y el input es el valor del dropdown. El usuario puede interactuar con dicho valor y hacerlo cambiar, cambiando así el input del callback y actualizando el output en consecuencia.

1. ‘Dash Documentation & User Guide | Plotly’ <https://dash.plotly.com/> [accessed 2 June 2022]. [↑](#endnote-ref-1)
2. ‘Dash HTML Components | Dash for Python Documentation | Plotly’ <https://dash.plotly.com/dash-html-components> [accessed 1 June 2022]. [↑](#endnote-ref-2)
3. ‘Dash Core Components | Dash for Python Documentation | Plotly’ <https://dash.plotly.com/dash-core-components> [accessed 29 March 2022]. [↑](#endnote-ref-3)
4. ‘Dash Bootstrap Components’ <https://dash-bootstrap-components.opensource.faculty.ai/> [accessed 2 June 2022]. [↑](#endnote-ref-4)
5. Aly Sivji, ‘Interactive, Web-Based Dashboards in Python’ <https://alysivji.github.io/./reactive-dashboards-with-dash.html> [accessed 26 May 2022]. [↑](#endnote-ref-5)
6. ‘Part 3. Basic Callbacks | Dash for Python Documentation | Plotly’ <https://dash.plotly.com/basic-callbacks> [accessed 26 May 2022]. [↑](#endnote-ref-6)