Clase 3 : Lab IV

1-creamos un componente "fila" (td) que la idea es devolver los <td> los cuales llamaremos con el nombre "app-fila", tenemos que hacer un input desde "datos.html" para pasarle el item de datos

\*se genera un problema ya que al mostrar la tabla dentro del "tr" hay "td" que están dentro de "app-fila" y para que no se escriba dentro de "app-fila" sino que solo directo devuelva los "td" dentro de “app-fila”, es decir que no retorne esta etiqueta (app-fila) sino solo su contenido.

Para que esto no suceda lo que hacemos es , vamos a "fila.td" y en “selector” le ponemos unos “[ ]” que esto lo que dice es "quiero que me devuelvas el contenido nada más"

 selector: '[app-fila]',

Entonces ahora devuelve solo el "td" sin el "app-fila" y desde "datos.html" al \*ngfor le tengo que poner 'app-fila [item]="item" ' dentro del <tr>

<tr \*ngFor="let item of datos" app-fila [item]="item">

Esto lo que hace es llamar al “componente” “app-fila” y al input “Item” le pasamos el ítem que obtenemos al iterar el array de “datos.

2-modificar el formulario para que cargue:

Nombre

Sexo(f | m | o)

Edad

Licencia (si | no)

Fecha nacimiento

3-Hay veces que vamos a querer hacer transformaciones sobre los datos que ingresen en los input, para hacer estas transformaciones:

Dentro de fila.html vamos a utilizar los "pipes" | los cuales nos permiten transformar datos, ejemplo el "datePipe".

Este pipe se puede usar tambien en "typescript"

Ejemplo : "item.fecha | date "

Podemos poner lo que querramos para el "|" y modificar los "standar" ejemplo "date: 'short' "

El "currency" muestra el signo $

El "lowerCase" pasa a minuscula

El "i18n" nos sirve para poner niestra pagina en distintas localizaciones digamos distintas las etiquetas dependiendo del lugar.

Podemos crear un custom pipe:

ng g pipe pipes/test

El método "transform" va a ser el que se va a encargar de la transformación del pipe,

Cuando creemos un pipe lo podremos utilizar en cualquier otro lugar

Crear un pipe y a otros ponerle lo que sea

\*Licencia , pintar de rojo o no (creamos uno propio)

Herramienta para comunicar entre hermano:

SERVICIOS

ng g service services/user

\*cuando lo creo no me hace el "update" y tenemos que hacerlo nosotros, vamos a "app.module.ts" y lo importamos y lo agregamos en "providers"

No es más que una clase el servicio , la cual nos va a servir , es una sola instacia (se carga una vez) si lo importamos en un componente y en otro , va a ser la misma instancia , por lo tanto como va a ser siempre la misma , vamos a modificarlo de un componente y verlo del otro.

Importamos el servicio en "saludar.ts" para usarlo ahí dentro, debemos inyectarlo en el constructor con el modificador de "accesibilidad" (obligatorio) user: UserService

Al servicio le cargamos el "listado"

Vamos a datos.ts , importamos el servicio , lo injertamos en el constructor