CoinSeeker

**Link:**

000webhost: <https://html5-css3-js-2020.000webhostapp.com/js-avanzado/index.html>

Firebase: <https://js-modulo-03.web.app/>

**Video reproductor personalizado**:

Aplicado en la sección principal: <https://html5-css3-js-2020.000webhostapp.com/js-avanzado/index.html>

|  |
| --- |
|  |
| Captura del reproductor de video: En naranja botones y en celeste animaciones con CSS |

* 1: Botón de retroceso, retrocede 6 segundos.
* 2: Botón detener.
* 3: Botón reproducir y pausa. Cuando el video esta pausado o detenido muestra el símbolo de reproducir, en cambio, cuando se esta reproduciendo muestra el símbolo de pausa.
* 4: Botón de adelantar, adelanta 6 segundos.
* 5: Animación aplicada en el hover y active. Se utiliza un SVG para poder modificar el color con propiedades de CSS.
* 6: Bordes redondeados con CSS.

Animación:

|  |
| --- |
|  |
| Captura de la animacion con CSS de la botonera del reproductor de video |

Funcionalidad:

Los cuatro botones ejecutan su tarea con el evento click. Los botones de avanzar y retroceder poseen una funcionalidad adicional: mientas se mantienen presionados se ejecuta el proceso de avance/retroceso de 6 segundos por cada medio segundo presionado (solo cuando se disparan los eventos mousedown y touchstart para evitar errores).

Para evitar errores/bug como que el video se siga adelantando/retrocediendo cuando se realiza el mousedown sobre el botón pero el mouseup fuera del mismo se añadió un addEventListener(“mouseleave”,…,…). De esta manera sin importar donde libere la presión de la tecla del mouse se da como finalizado el proceso de avance/retroceso continuo.

Para el responsive web se tuvo en cuenta este inconveniente, realizando la misma lógica, pero con los eventos touchstart, touchcancel y touchend.

|  |
| --- |
|  |
| Controlador del reproductor de video para botones avanzar y retroceder |

|  |
| --- |
|  |
| Controlador del reproductor de video para reproducir/pausa y detener |

Se visualiza en el código que al pausar o detener el video el tercer botón, de izquierda a derecha, automáticamente pasa a ser el de play. El botón vuelve a ser reasignado como pausa solo cuando el video se está reproduciendo.

|  |
| --- |
|  |
| eventListener utilizados y variables globales |

**Animaciones en otras secciones:**

En el footer:

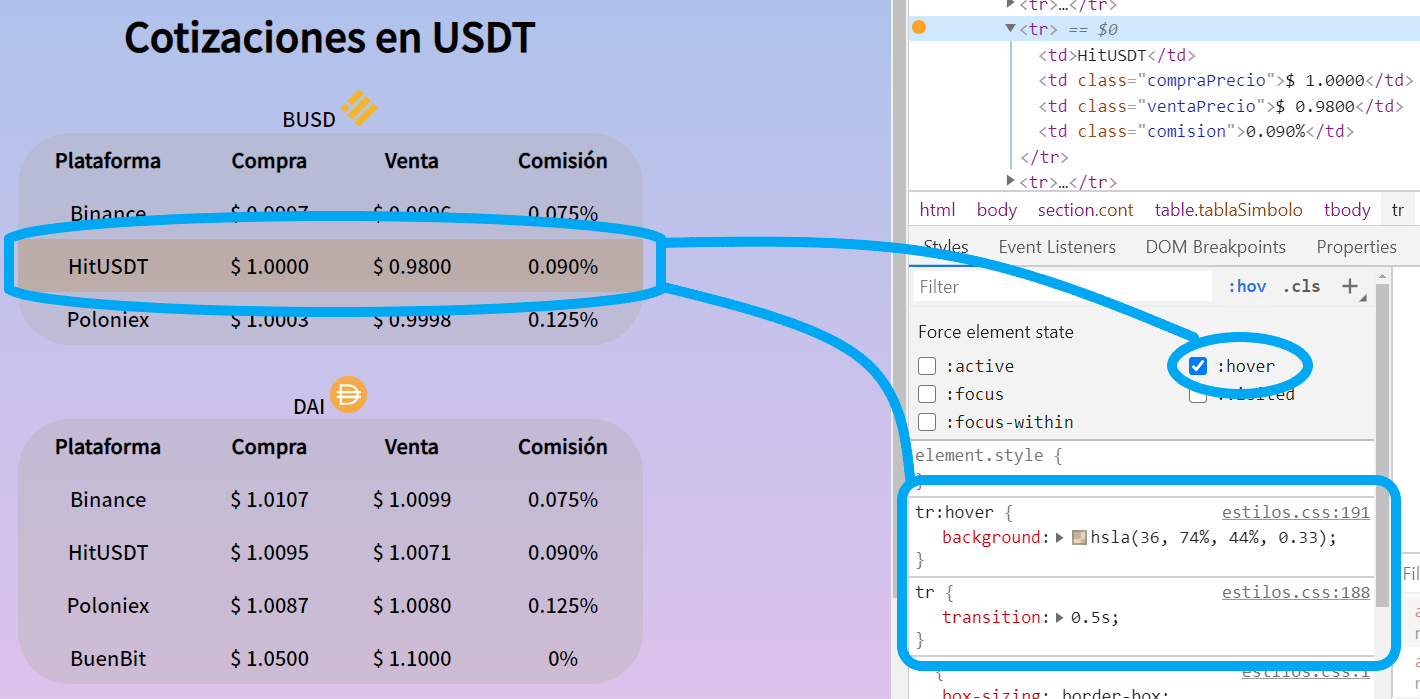
Se le incorpora animación de rotación de 360º y duración de 1.2 segundos a los cuatro iconos de las criptomonedas estables, al posicionarse sobre ellas.



|  |
| --- |
|  |
| Animación de los iconos del footer, giran 360º sin fin mientras se posiciones sobre ellos. |

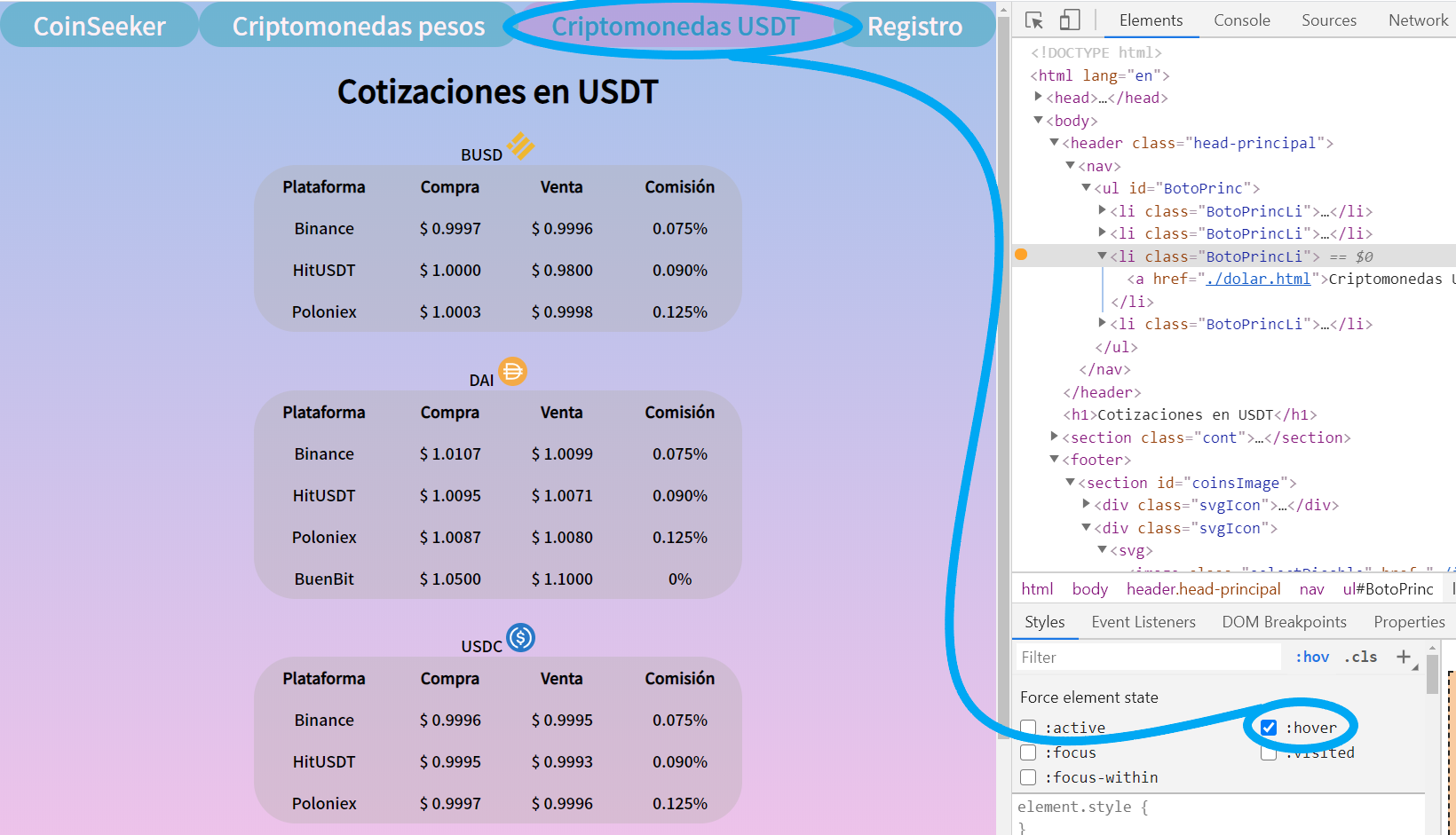
En las tablas:

Se realiza una animación en el hover sobre la fila posicionada, con una transición de 0.5 segundos.



En el header:

Se aplica una animación sobre las secciones del sitio en el header, con la propiedad animation-fill-mode: forwards para que sea un solo ciclo. El efecto de la animación es similar a la transición aplicada a la tabla, la idea en este es demostrar que hay mas de una manera de realizarlo.

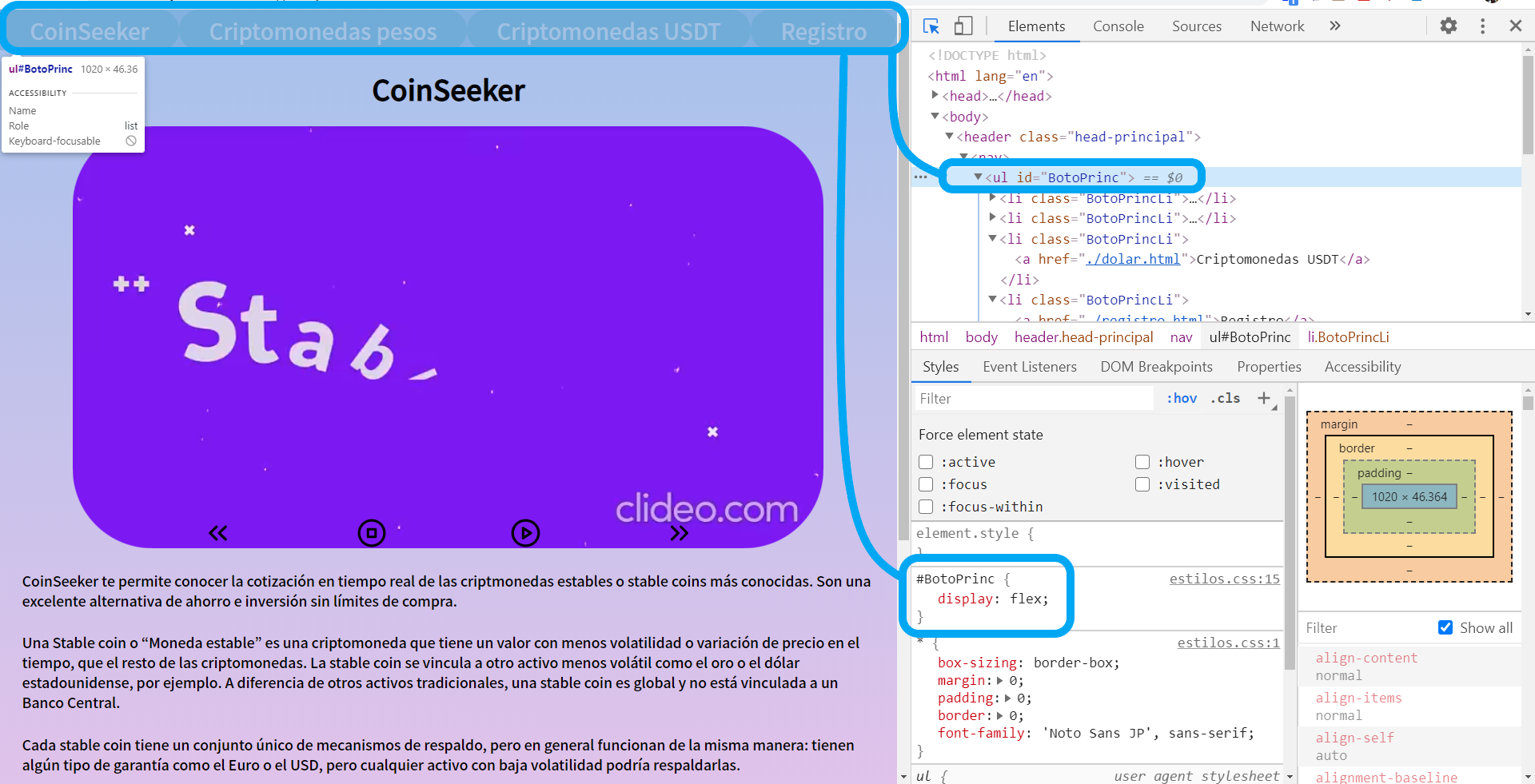


|  |
| --- |
|  |
| Animación de las diferentes secciones del sitio en el header |

**Flexbox:**

En el header:

Se aplica display:flex a la lista id=”botoPrinc” y flex-grow=1 a los flex-items para que ocupen todo el ancho disponible.



Ademas se vuelve a declarar a cada ítem de la lista como un nuevo contenedor flex para centrar fácilmente el elemento <a> en su interior.



Flexbox es aprovechado en el responsive web design, para cambiar el orden de los ítems de las listas, para el caso de pantalla mediana. Para pantalla chica el order vuelve a ser restablecido, ya que se utiliza flexbox para que quede uno por fila.

Para la pantalla mediana, con flex wrap=1 posibilito que los flex ítem se muestren en un nivel mas abajo.

Ya que resulta más prolijo que las secciones de cotizaciones estén al mismo nivel se cambia el orden de los flex ítem, inicialmente se ven así:



Pero al cambiar el orden se ven así:



Lista de Coinseeker order:1, Lista de Registro: order:2

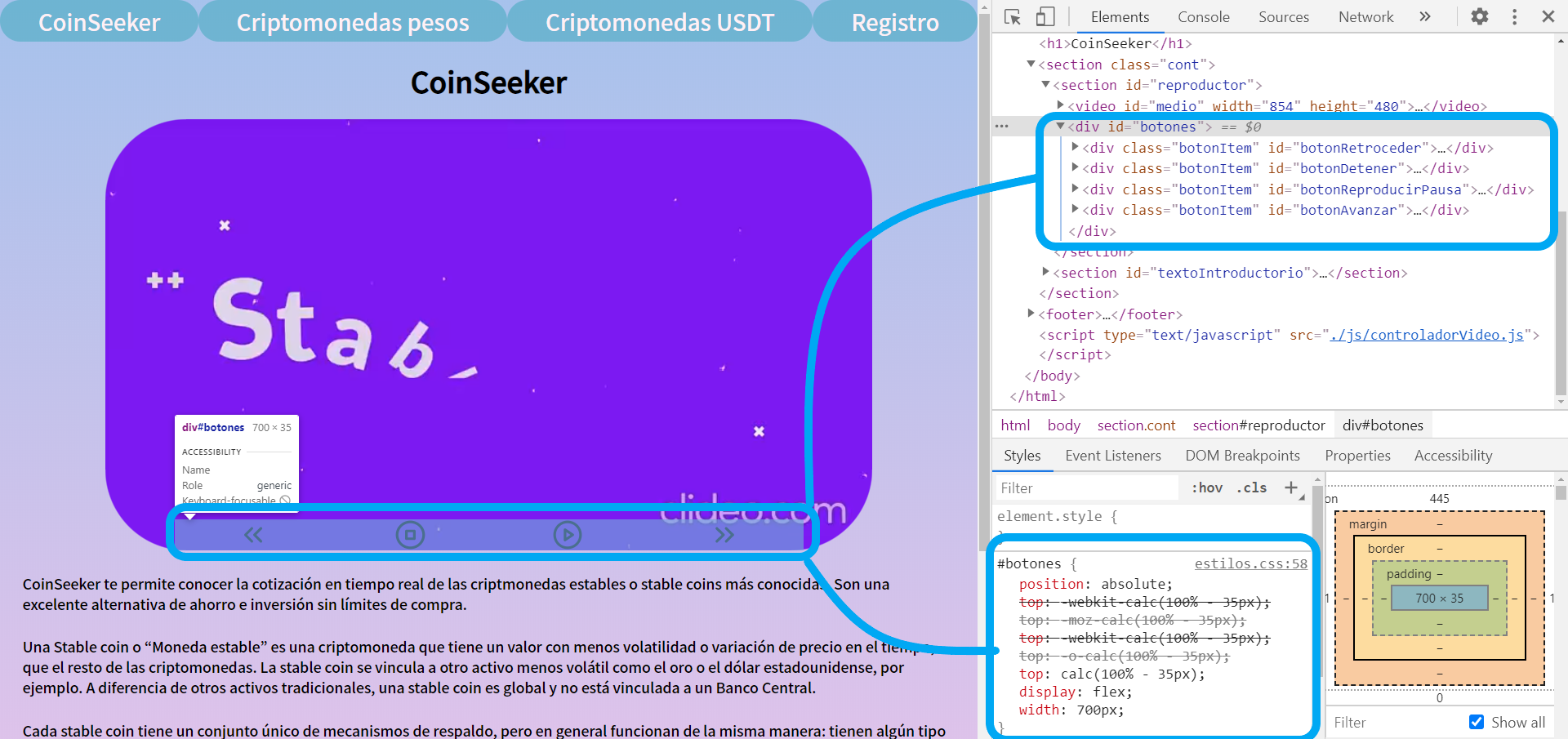
Lista de Criptomonedas pesos order:3, Lista de Criptomonedas USDT: order 4

Se utiliza la pseudoclase nth-child, además como parte de optimización de CSS.



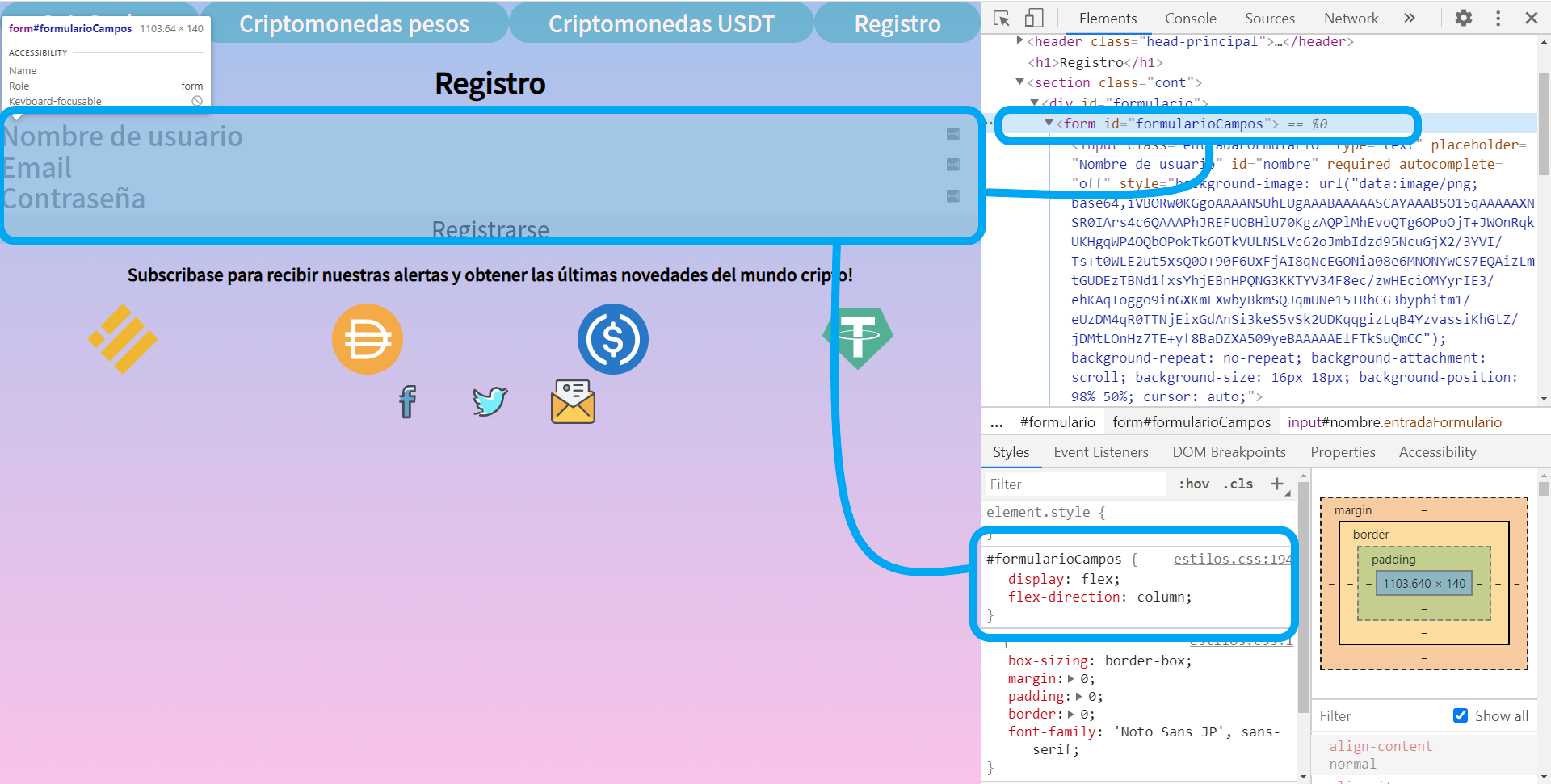
En el reproductor:

Es utilizado de manera simple para que cada botón ocupe el mismo ancho de manera fácil y rápida:



En el registro:

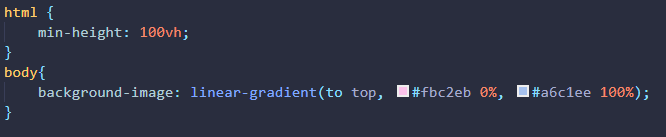
Se utiliza el principio que main-axis y el cross-axis pueden ser modificados, se modifica el main-axis para que tenga dirección de columna, es decir, vertical. De esta manera ya quedan todos los flex-item contenidos dentro del formularios alineados en vertical.



**Optimizaciones CSS:**

Además de las optimizaciones previamente mencionadas como los bordes redondeados, el cambio de color de los botones aprovechando una imagen con source tipo SVG, se aplicó lo siguiente:

* Fondo: Degradado del fondo, para que sea “infinito” se aplicó min-height: 100vh; para el html.



* Las tablas de cotización - pseudoclases: Para pantalla grande y mediana se leen en horizontal, pero para pantalla chica fue necesario hacerlo por columna. Para tal motivo el header de cada tabla debía desaparecer, pero para la información no sea mal interpretada se agrego las palabras compra, venta y comisión utilizando la pseudoclase :before.

|  |  |
| --- | --- |
|  |  |
| Tabla en pantalla grande y mediana | Tabla en pantalla chica |

|  |
| --- |
|  |
| Pseudoclase before aplicado en los elementos de clase “comision” |

* Las tablas de cotización – distinción de cada plataforma: Se señala cada plataforma con un borde inferior punteado negro. Al terminar la presentación de la información por cada plataforma, también de tabla, se lo indica con línea punteada, pero blanca.

Se redondean los bordes de solo la primera celda de la primera fila y la ultima celda de la ultima fila en la tabla.

