|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NOMBRE Y APELLIDOS:**  **Miguel Ramirez** | | | **FECHA: 09/08/2023** | | |
| **DOCENTE: MANUEL MACÍAS PÉREZ** | | | **NOTA:** | | |
| **(IFCD0210) DESARROLLO DE APLICACIONES CON TECNOLOGÍAS WEB.** | | | **Nº CURSO: 22-35/008902** | | |
| MF0491\_3 | UNIDADES DE APRENDIZAJE A LAS QUE RESPONDE: UA1 | |  | Duración: | 2h |
| UF1843 |
| PRÁCTICA FINAL |
| **DESCRIPCIÓN:**  El alumno de forma individual deberá realizar implementación de un programa utilizando un entorno integrado con las siguientes especificaciones para integrar técnicas de accesibilidad y usabilidad.  Según el contenido de un archivo en html:   * Crear estilos en un archivo css * Distribuir y mostrar su contenido según especificaciones de usabilidad utilizando JQuery. Utilizar los efectos de Jquery-ui. * Realizar pantallazo del resultado, adjuntarlo a este documento y convertir en PDF para enviar. * Subir todo el código a GitHub   **MEDIOS PARA SU REALIZACIÓN**   * Equipo microinformático y Software de base. * Herramientas y aplicaciones ofimáticas. * Aplicaciones de edición web. * Aplicación Visual Code Studio   **PAUTAS DE ACTUACIÓN DEL FORMADOR**  *Al inicio de la práctica, que se desarrollará de manera individual por cada uno de los alumnos, el formador/a realizará las siguientes actuaciones:*  - Fijará los objetivos de la práctica.  - Aportará las instrucciones necesarias a los alumnos/as para la realización de la misma, haciendo hincapié en aquellos aspectos más relevantes.  - Facilitará a cada alumno/a la documentación necesaria para el desarrollo de la práctica.  - Resolverá las dudas que se planteen durante el transcurso de la práctica, con objeto de que el alumnado aprenda y pueda concluir la realización de la misma.  Durante la realización de la práctica el formador/a supervisará el desarrollo de esta para evaluar tanto los procedimientos como el resultado final.  Al finalizar la práctica el formador examinará el desarrollo que han realizado los/as alumnos/as, proponiendo las medidas de corrección, en caso necesario.  ESPECIFICACIONES PARA LA EVALUACIÓN DE LA PRÁCTICA | | | | | |
| **Resultados a comprobar** | | **Indicadores de logro** | | | |
| 1. En un supuesto práctico, en el que se pide crear y mantener componentes software y documentos aplicar normas de accesibilidad y usabilidad para mejorar su utilización.  Conforme a los Criterios de evaluación CE1.3 | | Crea y archiva componentes software | | | |
| Modifica y elimina componentes software. | | | |
| Crea componentes software y documentos para aplicar normas de accesibilidad para mejorar su utilización. | | | |
| Mantiene componentes software y documentos aplicar para normas de accesibilidad para mejorar su utilización. | | | |
| Crea componentes software y documentos para aplicar normas de usabilidad para mejorar su utilización. | | | |
| Mantiene componentes software y documentos para aplicar normas de usabilidad para mejorar su utilización. | | | |

$(document).ready(function(){

    $("#dos").on("click", function(){

        $( ".contenedor" ).sortable();

        $( ".contenedor" ).disableSelection();

    });

    $("#tres").on("click", function(){

        $("#resizable").resizable({

            hadles: "se"

        });

    });

    $("#cuatro").on("click", function(){

        $("#selectable").selectable();

    });

    // Autocomplete

    $("#cinco").on("click", function(){1

        $.widget( "custom.combobox", {

            \_create: function() {

              this.wrapper = $( "<span>" )

                .addClass( "custom-combobox" )

                .insertAfter( this.element );

              this.element.hide();

              this.\_createAutocomplete();

              this.\_createShowAllButton();

            },

            \_createAutocomplete: function() {

              var selected = this.element.children( ":selected" ),

                value = selected.val() ? selected.text() : "";

              this.input = $( "<input>" )

                .appendTo( this.wrapper )

                .val( value )

                .attr( "title", "" )

                .addClass( "custom-combobox-input ui-widget ui-widget-content ui-state-default ui-corner-left" )

                .autocomplete({

                  delay: 0,

                  minLength: 0,

                  source: this.\_source.bind( this )

                })

                .tooltip({

                  classes: {

                    "ui-tooltip": "ui-state-highlight"

                  }

                });

              this.\_on( this.input, {

                autocompleteselect: function( event, ui ) {

                  ui.item.option.selected = true;

                  this.\_trigger( "select", event, {

                    item: ui.item.option

                  });

                },

                autocompletechange: "\_removeIfInvalid"

              });

            },

            \_createShowAllButton: function() {

              var input = this.input,

                wasOpen = false;

              $( "<a>" )

                .attr( "tabIndex", -1 )

                .attr( "title", "Show All Items" )

                .tooltip()

                .appendTo( this.wrapper )

                .button({

                  icons: {

                    primary: "ui-icon-triangle-1-s"

                  },

                  text: false

                })

                .removeClass( "ui-corner-all" )

                .addClass( "custom-combobox-toggle ui-corner-right" )

                .on( "mousedown", function() {

                  wasOpen = input.autocomplete( "widget" ).is( ":visible" );

                })

                .on( "click", function() {

                  input.trigger( "focus" );

                  // Close if already visible

                  if ( wasOpen ) {

                    return;

                  }

                  // Pass empty string as value to search for, displaying all results

                  input.autocomplete( "search", "" );

                });

            },

            \_source: function( request, response ) {

              var matcher = new RegExp( $.ui.autocomplete.escapeRegex(request.term), "i" );

              response( this.element.children( "option" ).map(function() {

                var text = $( this ).text();

                if ( this.value && ( !request.term || matcher.test(text) ) )

                  return {

                    label: text,

                    value: text,

                    option: this

                  };

              }) );

            },

            \_removeIfInvalid: function( event, ui ) {

              // Selected an item, nothing to do

              if ( ui.item ) {

                return;

              }

              // Search for a match (case-insensitive)

              var value = this.input.val(),

                valueLowerCase = value.toLowerCase(),

                valid = false;

              this.element.children( "option" ).each(function() {

                if ( $( this ).text().toLowerCase() === valueLowerCase ) {

                  this.selected = valid = true;

                  return false;

                }

              });

              // Found a match, nothing to do

              if ( valid ) {

                return;

              }

              // Remove invalid value

              this.input

                .val( "" )

                .attr( "title", value + " didn't match any item" )

                .tooltip( "open" );

              this.element.val( "" );

              this.\_delay(function() {

                this.input.tooltip( "close" ).attr( "title", "" );

              }, 2500 );

              this.input.autocomplete( "instance" ).term = "";

            },

            \_destroy: function() {

              this.wrapper.remove();

              this.element.show();

            }

          });

          $( "#combobox" ).combobox();

          $( "#toggle" ).on( "click", function() {

            $( "#combobox" ).toggle();

          });

    });

    $("#seis").on("click", function(){

        var state = true;

        $( "#button" ).on( "click", function() {

        if ( state ) {

            $( "#effect" ).animate({

            backgroundColor: "#aa0000",

            color: "#fff",

            width: 500

            }, 1000 );

        } else {

            $( "#effect" ).animate({

            backgroundColor: "#fff",

            color: "#000",

            width: 240

            }, 1000 );

        }

        state = !state;

        });

    });

    $("#siete").on("click", function(){

        if ( !$( "<canvas>" )[0].getContext ) {

            $( "<div>" ).text(

              "Your browser doesn't support canvas, which is required for this demo."

            ).appendTo( "#graphs" );

            return;

        }

        var i = 0,

        width = 100,

        height = 100;

      $.each( $.easing, function( name, impl ) {

        var graph = $( "<div>" ).addClass( "graph" ).appendTo( "#graphs" ),

          text = $( "<div>" ).text( ++i + ". " + name ).appendTo( graph ),

          wrap = $( "<div>" ).appendTo( graph ).css( 'overflow', 'hidden' ),

          canvas = $( "<canvas>" ).appendTo( wrap )[ 0 ];

        canvas.width = width;

        canvas.height = height;

        var drawHeight = height \* 0.8,

          cradius = 10,

          ctx = canvas.getContext( "2d" );

        ctx.fillStyle = "black";

        // Draw background

        ctx.beginPath();

        ctx.moveTo( cradius, 0 );

        ctx.quadraticCurveTo( 0, 0, 0, cradius );

        ctx.lineTo( 0, height - cradius );

        ctx.quadraticCurveTo( 0, height, cradius, height );

        ctx.lineTo( width - cradius, height );

        ctx.quadraticCurveTo( width, height, width, height - cradius );

        ctx.lineTo( width, 0 );

        ctx.lineTo( cradius, 0 );

        ctx.fill();

        // Draw bottom line

        ctx.strokeStyle = "#555";

        ctx.beginPath();

        ctx.moveTo( width \* 0.1, drawHeight + .5 );

        ctx.lineTo( width \* 0.9, drawHeight + .5 );

        ctx.stroke();

        // Draw top line

        ctx.strokeStyle = "#555";

        ctx.beginPath();

        ctx.moveTo( width \* 0.1, drawHeight \* .3 - .5 );

        ctx.lineTo( width \* 0.9, drawHeight \* .3 - .5 );

        ctx.stroke();

        // Plot easing

        ctx.strokeStyle = "white";

        ctx.beginPath();

        ctx.lineWidth = 2;

        ctx.moveTo( width \* 0.1, drawHeight );

        $.each( new Array( width ), function( position ) {

          var state = position / width,

            val = impl( state, position, 0, 1, width );

          ctx.lineTo( position \* 0.8 + width \* 0.1,

            drawHeight - drawHeight \* val \* 0.7 );

        });

        ctx.stroke();

        // Animate on click

        graph.on( "click", function() {

          wrap

            .animate( { height: "hide" }, 2000, name )

            .delay( 800 )

            .animate( { height: "show" }, 2000, name );

        });

        graph.width( width ).height( height + text.height() + 10 );

      });

    });

    $("#ocho").on("click", function(){

        // run the currently selected effect

        function runEffect() {

            // get effect type from

            var selectedEffect = $( "#effectTypes" ).val();

            // Most effect types need no options passed by default

            var options = {};

            // some effects have required parameters

            if ( selectedEffect === "scale" ) {

            options = { percent: 50 };

            } else if ( selectedEffect === "size" ) {

            options = { to: { width: 280, height: 185 } };

            }

            // Run the effect

            $( "#effect" ).show( selectedEffect, options, 500, callback );

        };

        //callback function to bring a hidden box back

        function callback() {

            setTimeout(function() {

            $( "#effect:visible" ).removeAttr( "style" ).fadeOut();

            }, 1000 );

        };

        // Set effect from select menu value

        $( "#button" ).on( "click", function() {

            runEffect();

        });

        $( "#effect" ).hide();

    });

    $("#nueve").on("click", function(){

        $( "#button" ).on( "click", function() {

            $( "#effect" ).removeClass( "newClass", 1000, callback );

          });

          function callback() {

            setTimeout(function() {

              $( "#effect" ).addClass( "newClass" );

            }, 3000 );

          }

    });

    $("#diez").on("click", function(){

        $( "#button" ).on( "click", function() {

            $( ".newClass" ).switchClass( "newClass", "anotherNewClass", 2000 );

            $( ".anotherNewClass" ).switchClass( "anotherNewClass", "newClass", 2000 );

        });

    });

    $("#once").on("click", function(){

        // run the currently selected effect

        function runEffect() {

            // get effect type from

            var selectedEffect = $( "#effectTypes" ).val();

            // Most effect types need no options passed by default

            var options = {};

            // some effects have required parameters

            if ( selectedEffect === "scale" ) {

            options = { percent: 50 };

            } else if ( selectedEffect === "size" ) {

            options = { to: { width: 200, height: 60 } };

            }

            // Run the effect

            $( "#effect" ).toggle( selectedEffect, options, 500 );

        };

        // Set effect from select menu value

        $( "#button" ).on( "click", function() {

            runEffect();

        });

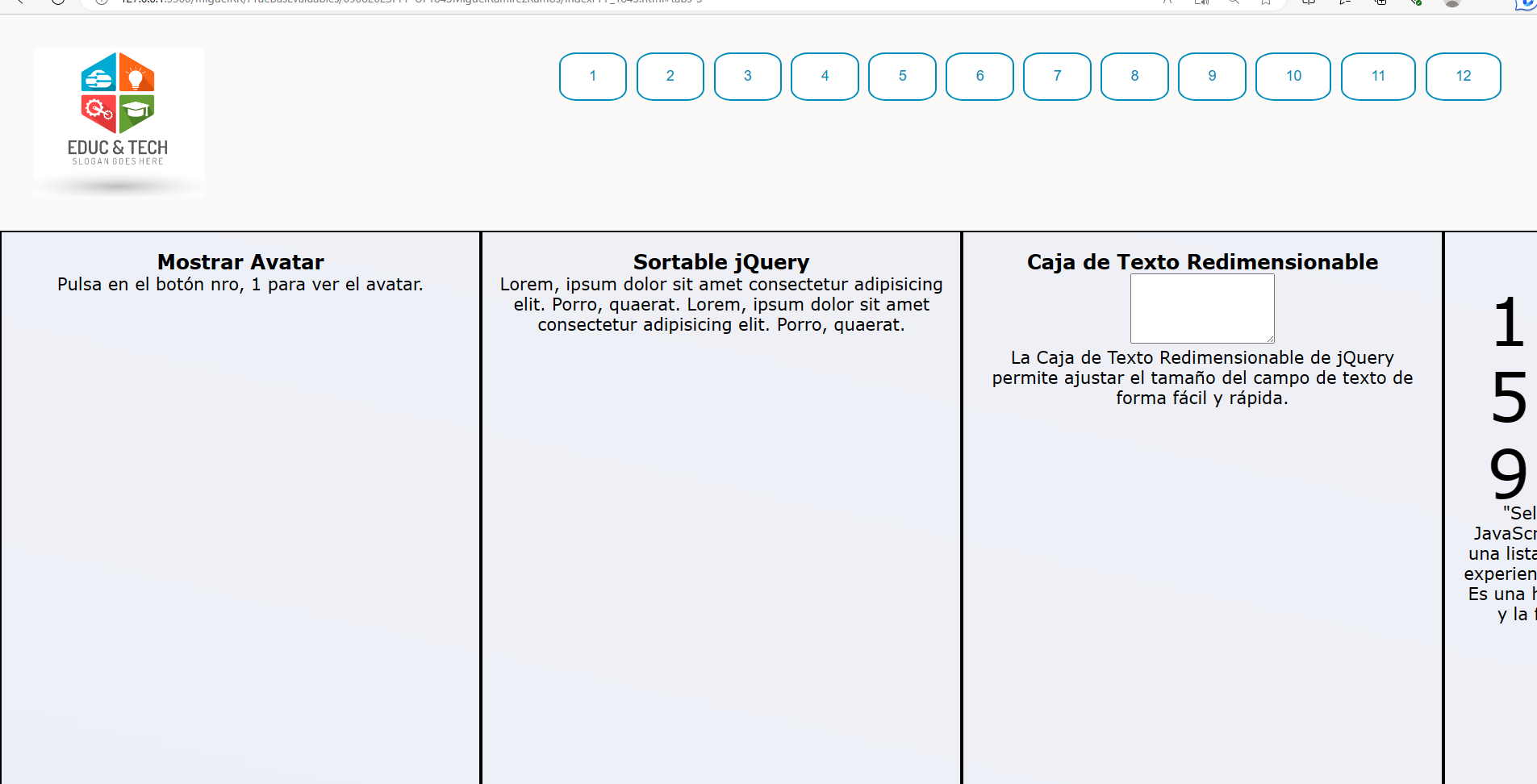
    });

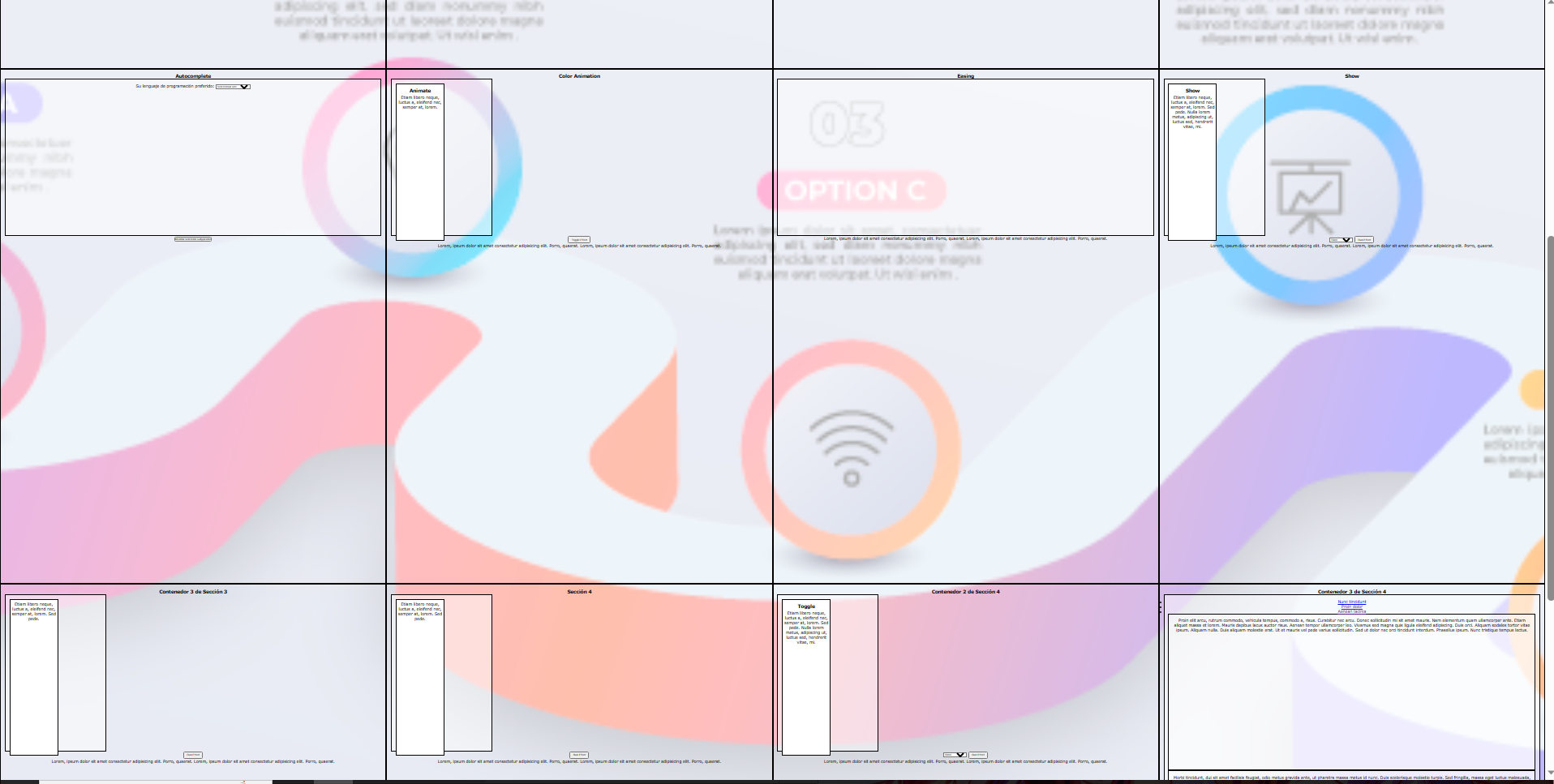
    $("#doce").on("click", function(){

        $( "#tabs" ).tabs();

    });

});

****

****

**Sistema de valoración**

Definición de indicadores y escalas de medida

Los indicadores que se van a establecer, será una hoja de chequeo, sistema de valoración, que complementa a este documento, donde se evalúan todos los resultados a comprobar (tareas). En este documento, se establecerán a su vez los indicadores de logro que se han de tener en cuenta, para conseguir los resultados a comprobar.

Mínimo exigible

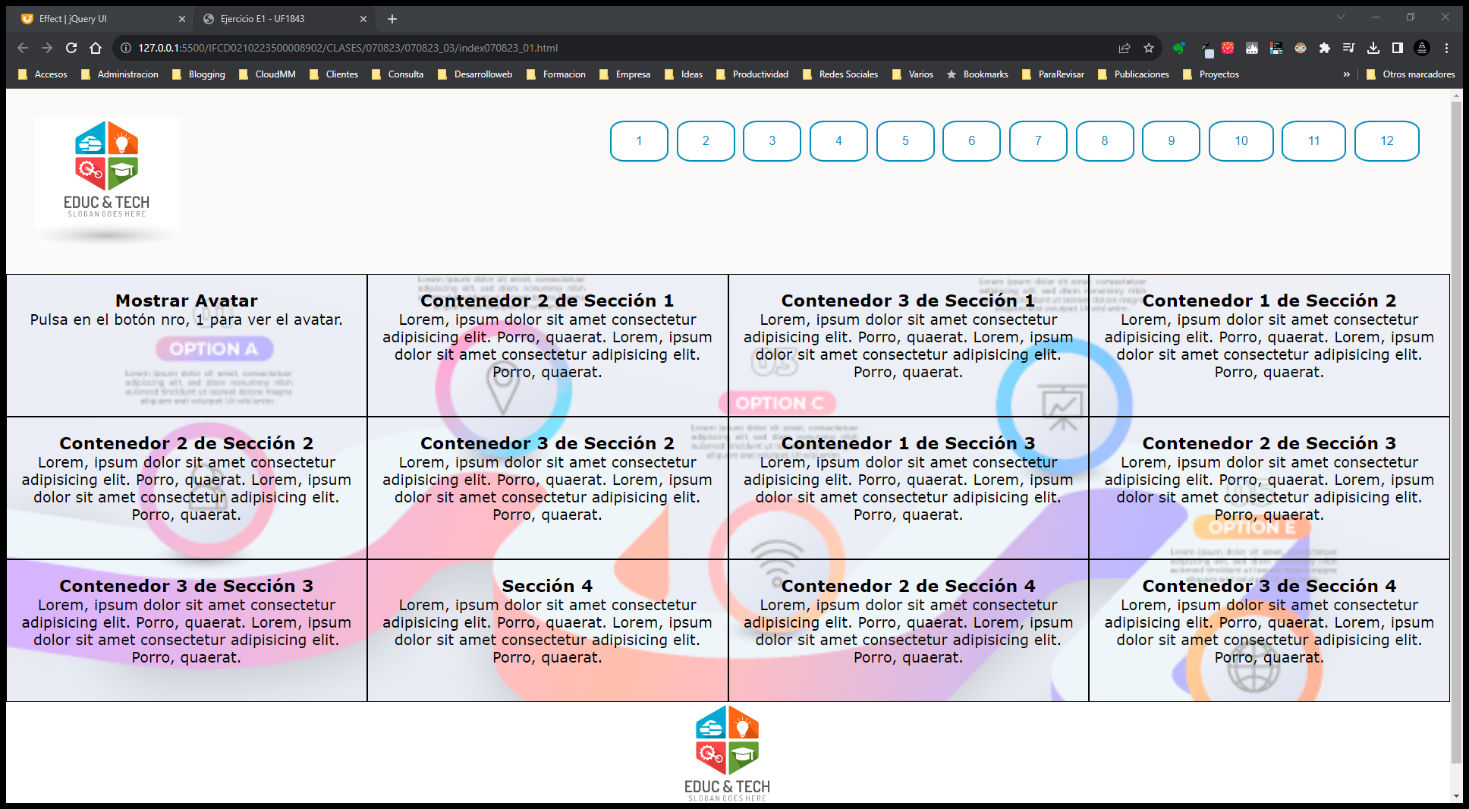
El mínimo exigible para la superación de la práctica es de 50 puntos sobre 100 puntos

**EJERCICIOS**

El alumno de forma individual deberá realizar implementación de un programa utilizando un entorno integrado con las siguientes especificaciones para integrar técnicas de accesibilidad y usabilidad.

Según el contenido de un archivo en html:

* Crear estilos en un archivo css
* Distribuir y mostrar su contenido según especificaciones de usabilidad utilizando JQuery. Utilizar los efectos de Jquery-ui.
* Realizar pantallazo del resultado, adjuntarlo a este documento y convertir en PDF para enviar.
* Subir todo el código a GitHub



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SISTEMA DE VALORACIÓN MF0491\_3- UF1843 – PRÁCTICA FINAL** | | | | |
| **NOMBRE DEL ALUMNO:** | | | | |
| **RESULTADO A COMPROBAR** | **INDICADORES DE LOGRO** | **ESCALA DE MEDIDA** | | |
| 1. En un supuesto práctico, en el que se pide crear y mantener componentes software y documentos aplicar normas de accesibilidad y usabilidad para mejorar su utilización.  Conforme a los Criterios de evaluación CE1.3 | Crea componentes software y documentos para aplicar normas de accesibilidad para mejorar su utilización. | - Crea componentes software y documentos para aplicar normas de accesibilidad para mejorar su utilización más del 75%  - Crea componentes software y documentos para aplicar normas de accesibilidad para mejorar su utilización entre un 50 % y 75%  - Crea componentes software y documentos para aplicar normas de accesibilidad para mejorar su utilización menos de un 50 % | B  R  M | 30  15  0 |
| Mantiene componentes software y documentos aplicar para normas de accesibilidad para mejorar su utilización. | - Mantiene componentes software y documentos aplicar para normas de accesibilidad para mejorar su utilización más del 75%  - Mantiene componentes software y documentos aplicar para normas de accesibilidad para mejorar su utilización entre un 50 % y 75%  - Mantiene componentes software y documentos aplicar para normas de accesibilidad para mejorar su utilización menos de un 50 % | B  R  M | 20  10  0 |
| Crea componentes software y documentos para aplicar normas de usabilidad para mejorar su utilización. | - Crea componentes software y documentos para aplicar normas de usabilidad para mejorar su utilización más del 75%  - Crea componentes software y documentos para aplicar normas de usabilidad para mejorar su utilización entre un 50 % y 75%  - Crea componentes software y documentos para aplicar normas de usabilidad para mejorar su utilización menos de un 50 % | B  R  M | 30  15  0 |
| Mantiene componentes software y documentos para aplicar normas de usabilidad para mejorar su utilización. | - Mantiene componentes software y documentos para aplicar normas de usabilidad para mejorar su utilización más del 75%  - Mantiene componentes software y documentos para aplicar normas de usabilidad para mejorar su utilización entre un 50 % y 75%  - Mantiene componentes software y documentos para aplicar normas de usabilidad para mejorar su utilización menos de un 50 % | B  R  M | 20  10  0 |
| Valor mínimo exigible: 50 | | Valor máximo: 100 |  | |