

Definición Entrega ET1

Interfaces de Usuario

Curso 2023-2024

Competencias a evaluar

Código	Descripción	ET1
A4	Coñecementos básicos sobre o uso e programación dos ordenadores, sistemas operativos, bases de datos e programas informáticos con aplicación na enxeñería	
A23	Capacidade para deseñar e avaliar interfaces persoa-computador que garantan a accesibilidade e usabilidade aos sistemas, servizos e aplicacións informáticas	X
A25	Capacidade para desenvolver, manter e avaliar servizos e sistemas software que satisfagan todos os requisitos do usuario e se comporten de forma fiable e eficiente, sexan asequibles de desenvolver e manter e cumpran normas de calidade, aplicando as teorías, principios, métodos e prácticas da Enxeñería do Software	X
A26	Capacidade para valorar as necesidades do cliente e especificar os requisitos software para satisfacer estas necesidades, reconciliando obxectivos en conflito mediante a procura de compromisos aceptables dentro das limitacións derivadas do custo, do tempo, da existencia de sistemas xa desenvolvidos e das propias organizacións	X
A28	Capacidade de identificar e analizar problemas e deseñar, desenvolver, implementar, verificar e documentar solucións software sobre a base dun coñecemento axeitado das teorías, modelos e técnicas actuais	X
A33	Capacidade para empregar metodoloxías centradas no usuario e a organización para o desenvolvemento, avaliación e xestión de aplicacións e sistemas baseados en tecnoloxías da información que aseguren a accesibilidade, ergonomía e usabilidade dos sistemas	
B1	Capacidade de análise, síntese e avaliación	X
B2	Capacidade de organización e planificación	X
B3	Comunicación oral e escrita na lingua nativa	
B5	Capacidade de abstracción: capacidade de crear e utilizar modelos que reflectan situacións reais	X
B8	Resolución de problemas	X
B9	Capacidade de tomar decisións	X
B10	Capacidade para argumentar e xustificar lóxicamente as decisións tomadas e as opinións	
B11	Capacidade de actuar autonomamente	X

B12	Capacidade de traballar en situacións de falta de información e/ou baixo presión	X
B13	Capacidade de integrarse rapidamente e traballar eficientemente en equipos unidisciplinares e de colaborar nun entorno multidisciplinar	
B15	Capacidade de relación interpersoal	
B16	Razoamento crítico	X
B18	Aprendizaxe autónoma	X
B19	Adaptación a novas situacións	
B20	Creatividade	X
B21	Liderazgo	
B22	Ter iniciativa e ser resolutivo	X

Tipología

Entrega individual de realización individual.

Definición

Dada la tabla que se proporciona en este punto y la definición de los formatos correctos de entrada de datos para cada uno de los atributos, se solicita la definición de los test posibles a realizar para cada campo y la batería de pruebas que verifica la prueba de los test definidos anteriormente.

--

-- Estructura de tabla para la tabla `programa`

--

```
CREATE TABLE `programa` (
  `id_programa` int(6) NOT NULL,
  `nombre_programa` varchar(60) NOT NULL,
  `acronimo_programa` varchar(20) NOT NULL,
  `nombre_original_programa` varchar(60) NOT NULL,
  `autor_programa` varchar(50) NOT NULL,
  `autor_original_programa` varchar(50) NOT NULL,
  `ano_programa` int(4) NOT NULL,
  `ano_original_programa` int(4) NOT NULL,
  `requisitos_programa` varchar(300) NOT NULL,
  `poblacion_desde_programa` int(2) NOT NULL,
  `poblacion_hasta_programa` int(2) NOT NULL,
  `unidad_poblacion` enum('MESES','AÑOS') NOT NULL,
  `tipo_programa` enum('EVALUACIÓN' , 'INTERVENCIÓN' , 'EVALUACIÓN E INTERVENCIÓN') NOT NULL,
  `tiempo_aplicacion_programa` int(4) NOT NULL COMMENT 'tiempo de aplicación de programa en min',
  `descrip_interp_programa` varchar(5000) NOT NULL COMMENT 'descripcion e interpretación del programa',
```

```

`fichero_programa` varchar(60) NOT NULL,
`enlace_programa` varchar(100) NOT NULL,
`formato_programa` enum('PAPEL','ELECTRÓNICO','PAPEL Y ELECTRÓNICO') NOT
NULL,
`modo_correccion_programa` enum('PAPEL','ELECTRÓNICO','PAPEL Y
ELECTRÓNICO') NOT NULL,
`modo_aplicacion_programa` enum('INDIVIDUAL','COLECTIVO','INDIVIDUAL Y
COLECTIVO') NOT NULL,
`imagen_programa` varchar(60) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Formatos permitidos

id_programa	dígitos min 1 max 6
nombre_programa	alfabéticos con acentos, ñ y espacios min 6 max 60
acronimo_programa	alfabéticos sin acentos ni espacios min 3 max 20
nombre_original_programa	alfabéticos con acentos, ñ y espacios min 6 max 60
autor_programa	alfabéticos con acentos, ñ y espacios min 6 max 50
autor_original_programa	alfabéticos con acentos, ñ y espacios min 6 max 50
ano_programa	dígitos 4 min 4 max, año no superior actual
ano_original_programa	dígitos 4 min 4 max, año no superior actual
requisitos_programa	alfabéticos con acentos, ñ, espacios y signos de puntuación min 6 max 300
poblacion_desde_programa	dígitos min 1 max 2
poblacion_hasta_programa	dígitos min 1 max 2
unidad_poblacion	solo valores 'MESES','AÑOS'
tipo_programa	solo valores 'EVALUACIÓN', 'INTERVENCIÓN', 'EVALUACIÓN E INTERVENCIÓN'
tiempo_aplicacion_programa	dígitos min 1 max 4
descrip_interp_programa	alfabéticos con acentos, ñ, espacios, signos de puntuación y retornos de carro min 100 max 5000

fichero_programa	alfabéticos sin acentos ni ñ ni espacios min 7 max 60
enlace_programa	alfabéticos y : y / y . sin acentos ni ñ ni espacios min 10 max 100
formato_programa	solo valores ('PAPEL', 'ELECTRÓNICO', 'PAPEL Y ELECTRÓNICO')
modo_correccion_programa	solo valores ('PAPEL', 'ELECTRÓNICO', 'PAPEL Y ELECTRÓNICO')
modo_aplicacion_programa	solo valores 'INDIVIDUAL', 'COLECTIVO', 'INDIVIDUAL Y COLECTIVO'
imagen_programa	alfabéticos sin acentos ni ñ ni espacios min 7 max 60

Objetivos

Lo que se pretende ahora es poder determinar todas las posibles pruebas de formato de datos introducidos en el formulario y sus mensajes correspondientes, que el usuario debe recibir como respuesta a la introducción de los mismos.

Se solicita:

- 1) Los datos de la entrega realizada, identificando la entrega, el alumno y el número de horas dedicadas.
- 2) Identificación de los test a realizar por cada campo del formulario con sus mensajes de respuesta.

Se especifica cada tipo de test que debería realizarse para cada campo para detectar los posibles errores de formato junto con el test para detectar el formato válido. **Es importante que las definiciones de test de error devuelvan un false en la prueba para comprobar si el valor es erróneo y devuelvan un true si el valor es válido. Solo la definición del test de formato válido debería responder un true.**

- 3) Pruebas con valores para cada test y mensajes de respuesta para validación de formato de cada campo.

Propósito

- 1) Realizar la entrega de un fichero texto ascii con el nombre

ET1_NombreApellidosAlumno.js para la indicación de los datos de entrega.

Debe contener la asignatura, la entrega, el nombre del alumno y las horas dedicadas en el total de la entrega con el siguiente formato:

datosgenerales_NombreApellidosAlumno = Array(Apellido1 Apellido2 Nombre (del Alumno), entrega, horasdedicadas);

Ejemplo:

datosgenerales_JavierRodeiroIglesias = Array('Rodeiro Iglesias Javier','ET1', 80);

1) Realizar la entrega de un fichero texto ascii con el nombre ET1_NombreApellidosAlumno_tests.js para la definición de los test de formato de campo.

Cada definición de test debe contener la siguiente información:

Campo

Número definición test (Consecutivo de 0-número de definiciones para cada campo)

Descripción test

Resultado esperado (true/false)

Mensaje respuesta

def_test_NombreApellidosAlumno =

Array(

//comentario informativo si se desea

Array(Campo,Número Descripción test,Descripción Test ,Resultado esperado ,Mensaje respuesta),

.

.

.

);

He aquí un ejemplo:

usuario : alfabético sin acentos ni ñ ni espacios entre 3 y 45 caracteres

Campo	Número Definición test	Descripción test	Resultado	Mensaje
usuario	1	Tamaño < 3	false	'El login del usuario no puede tener menos de 3 caracteres'
usuario	2	Tamaño > 45	false	'El login del

				usuario no puede tener más de 45 caracteres'
usuario	3	no alfabético o con acentos o con ñ o con espacios	false	'Login de usuario contiene caracteres no permitidos (solo alfabéticos sin acentos')
usuario	4	alfabético sin acentos ni ñ ni espacio	true	'Exito'

```

<script type="text/javascript">
def_test_JavierRodeirolglesias =
Array(
    //basicas de usuario
    Array('usuario',1,'tamaño < 3' ,false,'El login del usuario no puede tener menos de 3
caracteres'),
    Array('usuario',2,'tamaño > 45"',false,'El login del usuario no puede tener más de 45
caracteres' ),
    Array('usuario',3,'no alfabético o con acentos o con ñ o con espacios',false,'Login
de usuario contiene caracteres no permitidos (solo alfabéticos sin acentos)'),
    Array('usuario',4,'alfabético sin acentos ni ñ ni espacios',true,'Exito')
);
</script>

```

2) Realizar la entrega de un fichero texto ascii con el nombre ET1_NombreApellidosAlumno_pruebas.js para las pruebas con valores de los tests definidos en el documento anterior.

El fichero tendrá el siguiente formato:

```

pruebasunitarias_NombreApellidosAlumno = Array(
    //basicas de usuario
    Array(
        número definición de test, (conforme el fichero anterior)
        campo,
        número test, (consecutivo 0-total de pruebas realizadas para todos los
campos)
        valor,
        resultado(true/false)
    )
)

```

He aquí un ejemplo:

Número Definición test	campo	número test	valor	resultado
1	usuario	1	jr	false
1	usuario	2	javi	true
2	usuario	3	'aaaaaaaaaaaa aaaaaaaaaaaa aaaaaaaaaaaa aaaaaaaaaaaa aaaaaaaaaaaa aaa'	false
2	usuario	4	javi	true
3	usuario	5	j rodeiro	false
3	usuario	6	iñaki	false
3	usuario	7	javier	true
4	usuario	8	jose ford	false
4	usuario	9	jrodeiro	true

```
<script type="text/javascript">
pruebasunitarias_JavierRodeiroIglesias =
Array(
    //basicas de usuario
    Array(1,'usuario',1,'jr',false),
    Array(1,'usuario',2,'javi',true),
    Array(2,'usuario',3,'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaa',false),
    Array(2,'usuario',4,'javi',true),
    Array(3,'usuario',5,'j rodeiro',false),
    Array(3,'usuario',6,'iñaki',false),
    Array(3,'usuario',7,'javier',true),
    Array(4,'usuario',8,'jose ford',false),
    Array(4,'usuario',9,'jrodeiro',true)
);
</script>
```

Historias de usuario a cumplir

Particulares de entrega (Obligatorio. Si se incumple alguno de estos criterios la

nota será 0)

1. Los ficheros tienen el nombre, formato y tipo indicado en la entrega
2. El directorio a entregar existe y tiene el nombre indicado en la entrega
3. El alumno evaluado ha indicado el número de horas utilizadas en la realización de la entrega

Por cada error en la definición de test a realizar (p.e. el mensaje no es adecuado para la prueba, devuelve un true un test de error, devuelve un false un test de éxito, falta el test de éxito, falta algun test de error,.....) 0,1

Por cada error en la definición de pruebas de test a realizar (p.e., devuelve un true en una prueba de error, devuelve un false una prueba de éxito, falta la prueba de éxito, falta alguna prueba de error,.....) 0,1

Forma de entrega

1) En el ejercicio ET1 de moovi de la asignatura se introducirán un fichero ET1_NombreApellidosAlumno.rar. Para ello:

- a) Introducir los tres ficheros solicitados en un directorio con nombre ET1_NombreApellidosAlumno.
- b) Comprimir el directorio en formato rar y darle el nombre ET1_NombreApellidosAlumno.rar

(SE ENTREGA ANTES DEL DIA 15 DE OCTUBRE A LAS 23:59 HORAS)

Changelog

[1] xx/x/xxxx

- []texto original

por

- []cambio realizado

[2] yy/y/yyyy

- []texto original (en gestión de actividades)

por

- []cambio realizado