

Definición Entrega ET1

Interfaces de Usuario

Curso 2024-2025

Competencias a evaluar

A2 Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

A4 Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

B3 Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan.

B8 Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

B9 Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

C4 Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería

C23 Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas

C25 Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software

C26 Capacidad para valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas necesidades, reconciliando objetivos en conflicto mediante la búsqueda de compromisos aceptables dentro de las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones

C28 Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales

C33 Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas

D4 Capacidad de análisis, síntesis y evaluación

D5 Capacidad de organización y planificación

D6 Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales

D8 Capacidad de trabajar en situaciones de falta de información y/o bajo presión

D9 Capacidad de integrarse rápidamente y trabajar eficientemente en equipos unidisciplinarios y de colaborar en un entorno multidisciplinar

D10 Capacidad de relación interpersonal.

D11 Razonamiento crítico

D12 Liderazgo

Tipología

Entrega individual de realización individual.

Definición

Dadas las tablas que se proporcionan en este punto y la definición de los formatos correctos de entrada de datos para cada uno de los atributos, se solicitan los ficheros de 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 `analysis\_preparation`

--

```
CREATE TABLE `analysis_preparation` (  
  `id_analysis_preparation` int(11) NOT NULL,  
  `name_analysis_preparation` varchar(100) NOT NULL,  
  `description_analysis_preparation` varchar(5000) NOT NULL,  
  `bib_analysis_preparation` varchar(200) NOT NULL,  
  `file_analysis_preparation` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

Formatos permitidos

id_analysis_preparation	dígitos min 1 max 11
name_analysis_preparation	alfabéticos y espacios sin acentos ni ñ, min 8 max 100
description_analysis_preparation	alfabéticos y espacios sin acentos ni ñ, min 80 max 5000
bib_analysis_preparation	alfabéticos con acentos, ñ, espacios y signos de puntuación, min 6 max 200
file_analysis_preparation	alfabéticos sin acentos ni ñ ni espacios

	min 7 max 100. Solo pdf, doc o docx y tamaño de fichero menor de 2000000 bytes.
--	---

```
--
-- Estructura de tabla para la tabla `project`
--
```

```
CREATE TABLE `project` (
  `id_project` int(11) NOT NULL,
  `name_project` varchar(500) NOT NULL,
  `start_date_project` date NOT NULL,
  `end_date_project` date NOT NULL,
  `responsable_project` varchar(60) NOT NULL,
  `organization_project` varchar(100) NOT NULL,
  `description_project` varchar(500) NOT NULL,
  `file_project` varchar(100) NOT NULL,
  `code_project` varchar(50) NOT NULL,
  `acronym_project` varchar(15) NOT NULL,
  `id_sampling_methodology` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

Formatos permitidos

id_project	dígitos min 1 max 11
name_project	alfabéticos y espacios sin acentos ni ñ, min 15 max 100
start_date_project	fecha válida con formato dd/mm/aaaa
end_date_project	fecha válida con formato dd/mm/aaaa
responsable_project	alfabéticos con acentos, ñ y espacios min 6 max 60
organization_project	alfabéticos con acentos, ñ y espacios min 6 max 100
description_project	Cualquier ascii min 30 max 500
file_project	alfabéticos sin acentos ni ñ ni espacios min 7 max 100. Solo pdf, doc o docx y tamaño de fichero menor de 2000000 bytes.
code_project	alfabéticos con ñ, espacios y signos de puntuación, sin acentos, min 6 max 50

acronym_project	alfabéticos con ñ y signos de puntuación, sin acentos ni espacios, min 6 max 15
id_sampling_methodology	dígitos min 1 max 11

Objetivos

- Determinar la definición de los test que son necesarios para establecer la validez de los datos introducidos en el formularios y sus mensajes correspondientes, que el usuario debe recibir como respuesta a la introducción de los mismos.
- Determinar el conjunto de pruebas que se deben realizar para verificar el correcto funcionamiento de los test definidos tanto en su respuesta positiva como negativa.

Propósito

- Realizar la entrega de un un fichero texto ascii con el nombre ET1_NombreApellidosAlumno.js para la indicación de los datos de entrega.

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

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

Ejemplo:

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

- 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:

Entidad

Campo

Número definición test (Consecutivo de 0-número de definiciones totales)

Descripción test

Resultado esperado (true/false)

Codigo respuesta

Mensaje respuesta

```
def_test =  
Array(
```

//comentario informativo si se desea

Array(

Entidad,
Campo,
Número Descripción test,
Descripción Test,
Resultado esperado,
Codigo respuesta,
Mensaje respuesta),

.

.

.

);

He aquí un ejemplo:

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

Entidad	Campo	Número Definición test	Descripción n test	Resultado	Codigo respuesta	Mensaje
persona	usuario	1	Tamaño < 3	false	usuario_login_size_min_KO	'El login del usuario no puede tener menos de 3 caracteres',
persona	usuario	2	Tamaño > 45	false	usuario_login_size_max_KO	'El login del usuario no puede tener más de 45 caracteres',
persona	usuario	3	no alfabético o con acentos o con ñ o con espacios	false	usuario_login_format_KO	'Login de usuario contiene caracteres no permitidos (solo alfabéticos sin

						acentos')
persona	usuario	4	alfabético sin acentos ni ñ ni espacio	true	usuario_lo gin_OK	'Exito'

```
def_test =
Array(
    //basicas de usuario
    Array('persona','usuario',1,'tamaño < 3',false,'usuario_login_size_min_KO','El login
del usuario no puede tener menos de 3 caracteres'),
    Array('persona','usuario',2,'tamaño > 45"',false,'usuario_login_size_max_KO','El
login del usuario no puede tener más de 45 caracteres' ),
    Array('persona','usuario',3,'no alfabético o con acentos o con ñ o con
espacios',false,'usuario_login_format_KO','Login de usuario contiene caracteres no
permitidos (solo alfabéticos sin acentos)'),
    Array('persona','usuario',4,'alfabético sin acentos ni ñ ni
espacios',true,'usuario_login_OK','Exito')
);
```

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

He aquí un ejemplo:

Número Definición test	Entidad	campo	número test	valor	resultado

1	persona	usuario	1	jr	false
1	persona	usuario	2	javi	true
2	persona	usuario	3	'aaaaaaaaaaaa aaaaaaaaaaaa aaaaaaaaaaaa aaaaaaaaaaaa aaaaaaaaaaaa aaaa'	false
2	persona	usuario	4	javi	true
3	persona	usuario	5	j rodeiro	false
3	persona	usuario	6	iñaki	false
3	persona	usuario	7	javier	true
4	persona	usuario	8	jose ford	false
4	persona	usuario	9	jrodeiro	true

pruebasunitarias =

```
Array(
  //basicas de usuario
  Array(1,'persona','usuario',1,'jr',false),
  Array(1,'persona','usuario',2,'javi',true),
  Array(2,'persona','usuario',3,'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaaaa',false),
  Array(2,'persona','usuario',4,'javi',true),
  Array(3,'persona','usuario',5,'j rodeiro',false),
  Array(3,'persona','usuario',6,'iñaki',false),
  Array(3,'persona','usuario',7,'javier',true),
  Array(4,'persona','usuario',8,'jose ford',false),
  Array(4,'persona','usuario',9,'jrodeiro',true)
);
```

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

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 para validación de formato de cada campo.

Forma de entrega

1) Para entregar el ejercicio ET1 de moovi de la asignatura se procederá de la siguiente forma:

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 VIERNES DÍA 18 DE OCTUBRE A LAS 23:59 HORAS)