

ACTIVIDADES UD10 - SERVIDOR

Mediante las tareas que se detallan en este documento vamos a conseguir la parte de los resultados de aprendizaje cubiertos por la UD10, atendiendo a sus correspondientes criterios de evaluación:

RESULTADOS DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN
RA7. Desarrolla servicios web reutilizables y accesibles mediante protocolos web, verificando su funcionamiento.	<p>a) Se han reconocido las características propias y el ámbito de aplicación de los servicios web.</p> <p>b) Se han reconocido las ventajas de utilizar servicios web para proporcionar acceso a funcionalidades incorporadas a la lógica de negocio de una aplicación.</p> <p>c) Se han identificado las tecnologías y los protocolos implicados en el consumo de servicios web.</p> <p>d) Se han utilizado los estándares y arquitecturas más difundidos e implicados en el desarrollo de servicios web.</p> <p>e) Se ha programado un servicio web.</p> <p>f) Se ha verificado el funcionamiento del servicio web.</p> <p>h) Se ha documentado un servicio web.</p>

Consideraciones adicionales:

- Los CE asociados a la misma actividad tendrán la misma calificación, que será la resultante de la actividad total.
- Se ha de continuar comentando el código mediante el código de tarea. El código no documentado convenientemente no se considerará para su corrección.
- Cualquier copia (ya sea de fuentes externas, literal de los apuntes...) que no sea una respuesta original será calificada con 0. El código no estructurado, que presente dificultades para ser leído no podrá ser evaluado.

Actividad 1: Consumo de servicios web

Código: UD11.1	CE: RA7.a, b, c, d	IE: I3, I6	Puntos: 10	Estimación: 1.5 h
-----------------------	---------------------------	-------------------	-------------------	--------------------------

DESCRIPCIÓN

Documenta (con capturas de pantalla, explicaciones y posibles enlaces que hayas consultado) y justifica los pasos realizados.

a) **(4 puntos)** Utiliza el [espacio OAuth 2](#) para probar al servicio Cloud Text-to-Speech API v1 (es el scope que has de introducir en el Step 1). Consultando la documentación de [Cloud Text-to-Speech API v1](#), realiza una petición que liste las voces que se pueden utilizar para sintetizar voz a partir de texto.

b) **(6 puntos)** En este apartado vamos a utilizar la API de Github para realizar una consulta tras habernos autenticado. La consulta consiste en recuperar los repositorios más valorados en Github (los que tengan más de 5000 estrellas) para el lenguaje Python, ordenándolos descendientemente por el número de estrellas. Deberás tener en cuenta las siguientes consideraciones:

- El endpoint a utilizar es: <https://api.github.com/search/repositories>, y puedes encontrar su descripción [aquí](#). Además, es importante conocer los [cualificadores](#) de búsqueda para poder filtrar la consulta por el número de estrellas.
- Previamente deberás crear una cuenta en Github, con tu e-mail de alu.edu.gva.es
- Según se detalla en [documentación](#) del API REST, se permite una mayor tasa de consultas si existe autenticación previa. Por tanto, vamos a utilizar autenticación tipo basic para hacer la consulta.
- Para la autenticación tipo basic, se ha de obtener previamente un token de Github. Sigue este [enlace](#) para obtener un token personal. Tendrás que utilizar este token como password cuando quieras realizar la consulta.
- El usuario (e-mail de Github) y el token de acceso se enviarán en la propia consulta al API de Github.

Actividad 2: Serializadores

Código: UD11.2	CE: RA7.e (20%)	IE: I3, I6	Puntos: 10	Estimación: 2h
-----------------------	------------------------	-------------------	-------------------	-----------------------

DESCRIPCIÓN

Crea los serializadores del Anexo I de este documento, con las características que se especifican.

Actividad 3: Vistas y URLs

Código: UD11.3	CE: RA7.e (80%)	IE: I3, I6	Puntos: 10	Estimación: 5h
-----------------------	------------------------	-------------------	-------------------	-----------------------

DESCRIPCIÓN

Realiza las siguientes tareas:

a) Crea las vistas y URLs que se detallan en el Anexo II de este documento, con las

características que se detallan. **(4 puntos)**

b) Para todas las vistas CRUD con opción a borrado, hay que mostrar un mensaje de error amigable (no asociado a ningún campo) cuando existen dependencias: "No se puede realizar la operación de borrado porque existen dependencias.". Crear un mixin para ello. **(0.5 puntos)**

c) Al crear un nuevo alumno mediante la vista AlumnoDetailViewSet, se ha de generar la información de ese alumno en CalificaciónUDCE, sustituyendo, de esta forma, la señal programada en la UD9 - Actividad 5. **(0.5 puntos)**

d) Programar la vista CalcularNotasView (basándote en el punto 4.4.11 del documento con el ejemplo guiado), que ha de calcular la nota total de un alumno por cada módulo, y rellenar secuencialmente las calificaciones en CalificacionCE, CalificacionRA, CalificacionTotal, con las siguientes variaciones: **(5 puntos)**

- Podrá recibir como parámetro el ID de un alumno y/o un ID de módulo.
- Si no recibe ningún parámetro, realizará los cálculos para todos los alumnos.
- Si recibe un ID de módulo, realizará los cálculos para todos los alumnos del módulo.
- Si recibe un ID de módulo y un ID de alumno, realizará los cálculos para ese alumno en ese módulo.
- Antes del inicio de los cálculos, se limpiarán los datos (CalificacionCE, CalificacionRA, CalificacionTotal) para ese alumno en particular.

Actividad 4: Documentación

Código: UD7.4	CE: RA7.h	IE: I3, I6	Puntos: 10	Estimación: 0.5h
----------------------	------------------	-------------------	-------------------	-------------------------

DESCRIPCIÓN

Genera el docstring de cada vista como se detalla en este [enlace](#), aportando una descripción significativa del propósito de cada vista, y su funcionamiento.

Actividad 5: Verificación

Código: UD7.5	CE: RA7.f	IE: I3, I6	Puntos: 10	Estimación: 0.5h
----------------------	------------------	-------------------	-------------------	-------------------------

DESCRIPCIÓN

Crea un documento, adjunto a estas actividades, donde pongas una captura de pantalla de los resultados de cada vista (del API navegable de Django), junto con sus información (métodos, etc.). Especifica el nombre de la vista, y a continuación su captura.

ANEXO I: Definición de serializadores

Nombre	app	Campos
ModuloListSerializer	core	- id - Representación textual del objeto.
ModuloDetailSerializer	core	Todos los del modelo.
RAListSerializer	core	- id - Representación textual del objeto.
RADetailSerializer	core	Todos los del modelo.
CEListSerializer	core	- id - minimo: valores Sí/No - Representación textual del objeto.
CEDetailSerializer	core	Todos los del modelo.
UnidadListSerializer	prog_didactica	- id - Representación textual del objeto.
UnidadDetailSerializer	prog_didactica	Todos los del modelo.
IEListSerializer	prog_didactica	- id - Representación textual del objeto.
IEDetailSerializer	prog_didactica	Todos los del modelo.
PondRAListSerializer	prog_didactica	- id - Representación textual del objeto.
PondRADetailSerializer	prog_didactica	Todos los del modelo.
PondCEListSerializer	prog_didactica	- id - Representación textual del objeto.
PondCEDetailSerializer	prog_didactica	Todos los del modelo.
PondCEUDListSerializer	prog_didactica	- id - Representación textual del objeto.
PondCEUDDetailSerializer	prog_didactica	Todos los del modelo.
AlumnoListSerializer	prog_aula	- id - Representación textual del objeto.
AlumnoDetailSerializer	prog_aula	Todos los del modelo.

CEUDListSerializer	prog_aula	- id - Representación textual del objeto.
CEUDDetailSerializer	prog_aula	Todos los del modelo.
CalUDCEListSerializer	prog_aula	- id - Representación textual del objeto.
CalUDCEDetailSerializer	prog_aula	Todos los del modelo.
CalCEListSerializer	prog_aula	- id - Representación textual del objeto.
CalCEDetailSerializer	prog_aula	Todos los del modelo.
CalRALListSerializer	prog_aula	- id - Representación textual del objeto.
CalRADetailSerializer	prog_aula	Todos los del modelo.
CalTotalListSerializer	prog_aula	- id - Representación textual del objeto.
CalTotalDetailSerializer	prog_aula	Todos los del modelo.

ANEXO II: Definición de vistas y URLs

- Definir cada una en su app correspondiente.
- Los queryset de las vistas han de recoger todos los posibles objetos de cada modelo.

Nombre	Serializador	Características
ModuloListViewSet	ModuloListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado. - Ordenación: codigo (defecto), nombre - Búsqueda: codigo, nombre - Paginación: Sin paginar.
ModuloDetailViewSet	ModuloDetailSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: CRUD
RAListViewSet	RAListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado. - Ordenación: codigo (defecto), nombre - Búsqueda: codigo, descripcion - Filtrado: modulo - Paginación: Paginado, tamaño standard.
RADetailViewSet	RADetailSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: CRUD
CEListViewSet	CEListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado (str, mínimo). - Ordenación: codigo (defecto), descripcion - Búsqueda: codigo, descripcion, RA.codigo, RA.descripcion, modulo.codigo, modulo.nombre - Filtrado: modulo, RA - Paginación: Paginado, tamaño standard.
CEDetailViewSet	CEDetailSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: CRUD
UnidadListViewSet	UnidadListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado. - Ordenación: nombre (defecto), - Búsqueda: nombre - Filtrado: sin filtrar. - Paginación: Sin paginar.
UnidadDetailViewSet	UnidadDetailSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: CRUD
IEListViewSet	IEListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado. - Ordenación: codigo (defecto), nombre - Búsqueda: codigo, nombre, descripcion

		<ul style="list-style-type: none"> - Filtrado: sin filtrar. - Paginación: Sin paginar.
IEDetailViewSet	IEDetailSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: CRUD
PondRAListViewSet	PondRAListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado. - Ordenación: combinación de modulo.nombre + RA.codigo (defecto), RA.nombre - Búsqueda: modulo.nombre, RA.codigo, RA.nombre - Filtrado: modulo - Paginación: Paginado, tamaño standard.
PondRADetailViewSet	PondRADetailSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: CRUD - Operaciones adicionales: implementar la validación de la UD7, Actividad 2, tarea g.
PondCEListViewSet	PondCEListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado. - Ordenación: combinación de modulo.nombre + RA.codigo + CE.codigo (defecto), CE.nombre - Búsqueda: modulo.nombre, RA.codigo, RA.nombre, CE.codigo, CE.nombre - Filtrado: modulo, RA - Paginación: Paginado, tamaño standard.
PondCEDetailViewSet	PondCEDetailSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: CRUD - Operaciones adicionales: implementar la validación de la UD7, Actividad 2, tarea h.
PondCEUDListViewSet	PondCEUDListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado. - Ordenación: combinación de unidad.nombre + modulo.nombre + RA.codigo + CE.codigo (defecto), CE.nombre - Búsqueda: modulo.nombre, RA.codigo, RA.nombre, CE.codigo, CE.nombre, unidad.nombre - Filtrado: modulo, RA, CE, unidad - Paginación: Paginado, tamaño standard.
PondCEUDDetailViewSet	PondCEUDDetailSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: CRUD - Operaciones adicionales: implementar la validación de la UD7, Actividad 2, tarea i.
AlumnoListViewSet	AlumnoListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado. - Ordenación: apellidos + nombre (defecto), ciudad, codigo_postal - Búsqueda: nombre, apellidos, direccion, codigo_postal, ciudad

		<ul style="list-style-type: none"> - Filtrado: sin filtrar - Paginación: Paginado, tamaño standard.
AlumnoDetailViewSet	AlumnoDetailSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: CRUD
CEUDListViewSet	CEUDListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado. - Ordenación: combinación de modulo.nombre + RA.codigo + CE.codigo + unidad.nombre(defecto), CE.codigo, CE.nombre - Búsqueda: modulo.nombre, RA.codigo, RA.nombre, CE.codigo, CE.nombre, unidad.nombre - Filtrado: modulo, RA, CE, unidad - Paginación: Paginado, tamaño standard.
CEUDDetailViewSet	CEUDDetailSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: CRUD
CalUDCELListViewSet	CalUDCELListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado. - Ordenación: combinación de unidad.nombre + modulo.nombre + RA.codigo + CE.codigo + alumno.apellidos + alumno.nombre (defecto), CE.codigo, CE.nombre - Búsqueda: modulo.nombre, RA.codigo, RA.nombre, CE.codigo, CE.nombre, unidad.nombre, alumno.nombre, alumno.apellidos - Filtrado: alumno, modulo, RA, CE, unidad - Paginación: Paginado, tamaño large.
CalUDCEDetailViewSet	CalUDCEDetailSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: CRUD
CalCELListViewSet	CalCELListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado. - Ordenación: combinación de modulo.nombre + RA.codigo + CE.codigo + alumno.apellidos + alumno.nombre (defecto), CE.codigo, CE.nombre - Búsqueda: modulo.nombre, RA.codigo, RA.nombre, CE.codigo, CE.nombre, alumno.nombre, alumno.apellidos - Filtrado: alumno, modulo, RA, CE - Paginación: Paginado, tamaño large.
CalCEDetailViewSet	CalCEDetailSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: CRUD
CalRALListViewSet	CalRALListSerializer	<ul style="list-style-type: none"> - Operaciones CRUD: listado. - Ordenación: combinación de modulo.nombre + RA.codigo + alumno.apellidos + alumno.nombre,

		RA.codigo, RA.nombre - Búsqueda: modulo.nombre, RA.codigo, RA.nombre, alumno.apellidos - Filtrado: alumno, modulo, RA - Paginación: Paginado, tamaño standard .
CalRADetailViewSet	CalRADetailSerializer	- Operaciones CRUD: CRUD
CalTotalListViewSet	CalTotalListSerializer	- Operaciones CRUD: listado. - Ordenación: combinación de modulo.nombre + alumno.apellidos + alumno.nombre - Búsqueda: modulo.nombre, alumno.apellidos - Filtrado: alumno, modulo - Paginación: Paginado, tamaño standard .
CalTotalDetailViewSet	CalTotalDetailSerializer	- Operaciones CRUD: CRUD
CalcularNotasView	-	Ver actividad 3, tarea d), con instrucciones adicionales.

URLs

Vista	basename/name
ModuloListViewSet	modulo_list
ModuloDetailViewSet	modulo_detail
RAListViewSet	ra_list
RADetailViewSet	ra_detail
CEListViewSet	ce_list
CEDetailViewSet	ce_detail
UnidadListViewSet	unidad_list
UnidadDetailViewSet	unidad_detail
IELListViewSet	ie_list
IEDetailViewSet	ie_detail
PondRALListViewSet	pond_ra_list
PondRADetailViewSet	pond_ra_detail

PondCEListViewSet	pond_ce_list
PondCEDetailViewSet	pond_ce_detail
PondCEUDListViewSet	pond_ce_ud_list
PondCEUDDetailViewSet	pond_ce_ud_detail
AlumnoListViewSet	alumno_list
AlumnoDetailViewSet	alumno_detail
CEUDListViewSet	ce_ud_list
CEUDDetailViewSet	ce_ud_detail
CalUDCEListViewSet	cal_ud_ce_list
CalUDCEDetailViewSet	cal_ud_ce_detail
CalCEListViewSet	cal_ce_list
CalCEDetailViewSet	cal_ce_detail
CalRALListViewSet	cal_ra_list
CalRADetailViewSet	cal_ra_detail
CalTotalListViewSet	cal_total_list
CalTotalDetailViewSet	cal_total_detail
CalcularNotasView	calcular_notas