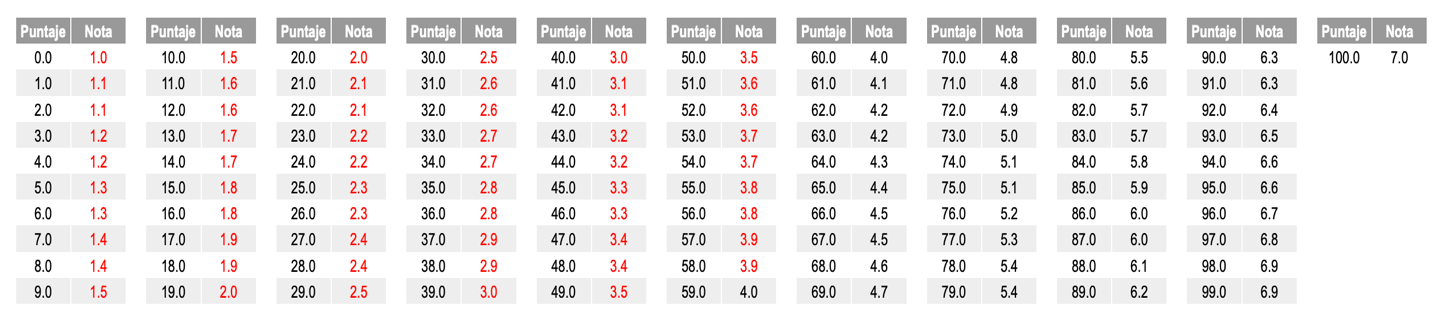
Evaluación Sumativa N°4: API RESTful con Integración de Base de Datos. (30%)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ÁREA ACADÉMICA** | | Tecnologías De Información Y Ciberseguridad | | | **CARRERA** | Analista programación | |
| **ASIGNATURA** | Programación Backend | | | | | **CÓDIGO** | TI2041 |
| **SEDE** | Talca | | | **DOCENTE** | Cristofher Rojas | | |
| **Unidad de Aprendizaje** | | | N° 4 | **Criterios a Evaluar** | | 3.1.1 – 3.1.2 – 3.1.3 – 3.1-4 | |
| **DURACIÓN** | 24 horas | | | **FECHA** | | 19-12-2024 | |

|  |
| --- |
| **Unidad de Aprendizaje**  1.- API RESTful con Integración de Base de Datos.  **Aprendizaje esperado**  Construye API RESTFul usando como autenticación JWT, según requerimiento. (Integrada Competencia Genérica Trabajo en Equipo, Nivel 1) |

|  |
| --- |
| **Criterios de evaluación**  3.1.1.- Implementa DjangoRest Framework según requerimiento.  3.1.2.- Codifica instrucciones de autenticación según requerimiento  3.1.3.- Codifica instrucciones que generen salidas en formato JSON según requerimiento.  3.1.4.- Cumple con las tareas asignadas en el tiempo definido.  3.1.5.- Aporta información e ideas para la consecución de las tareas.  **Contenidos:**   * Concepto Restful API. * Arquitectura REST. * Conexión a una base de datos a través del Framework. * Operaciones CRUD a través del Framework. * Seguridad |

|  |
| --- |
| **INSTRUCCIONES GENERALES:**   1. La evaluación es de carácter grupal de un máximo de 2 personas. 2. La nota 4.0 se obtiene logrando un 60% del puntaje total. 3. La evaluación contempla un total de 100 puntos, requiriendo 60 puntos para el 4,0. 4. Preocúpese de la redacción y ortografía de sus respuestas. 5. Durante el horario de clase es posible realizar consultas (sólo del enunciado de la prueba), después de dicho lapso, no se aceptarán preguntas de ningún tipo. 6. Cualquier copia que se detecte en respuestas de una o más evaluaciones será sancionada según lo indicado en el Reglamento Académico del estudiante. |

****

**Indicaciones generales:**

La presente evaluación implica la realización de una serie de actividades, de manera grupal, donde los estudiantes en base a un problema planteado por el docente acerca del desarrollo de una solución informática realizan un levantamiento de requerimientos empleando para ello conocimientos y habilidades relacionadas con la programación back end, para así lograr dar respuesta a las necesidades que son presentadas.

**Actividades:**

1. Un famoso local gastronómico de la región, les solicita realizar la siguiente aplicación web:
   1. La aplicación web necesita una página principal (index o landing page), donde se presenta la información básica de la empresa. Se puede utilizar algún template de bootstraps para la página de presentación.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* 1. Se desea implementar un sistema web, que permita almacenar reservas de mesas del local gastronómico.
  2. El sistema debe permitir desplegar el listado de reservas disponibles, además de poder agregar, modificar y eliminar reservas (CRUD).
  3. Para cada Reserva se necesita almacenar, su ID, nombre de la persona que realiza la reserva, teléfono, la fecha de reserva, la hora, la cantidad de personas, el estado (RESERVADO, COMPLETADA, ANULADA, NO ASISTEN) y una observación.

1. Se debe configurar el administrador por defecto de Django, el usuario debe ser “admin” y su contraseña “secret”, se deben poder administrar las reservas igualmente y debe estar en español. (Deben adjuntar la base de datos)
2. Todos los campos son obligatorios, excepto la observación, la cantidad de personas debe estar entre 1 y 15.
3. La información debe ser almacenada en una BD MySQL llamada “DJANGO\_RESERVAS”, debe generar las migraciones en su proyecto.
4. El sistema debe contar con una API publica que permita manipular la información pues se desea a futuro realizar una aplicación móvil para la administración.
   1. La API REST debe ser realizada mediante Class Based Views o Función Based Views y debe permitir listar todas las reservas ordenadas por fecha, buscar una en particular (dado su id), agregar, modificar y eliminar.

• El proyecto debe llamarse EVA4\_FINAL

• Debe utilizar Django y Django Rest Framework para el proyecto

• Puede utilizar algún framework como Bootstrap para la interfaz o aplicar sus diseños con css de forma manual.

* Cronograma para el desarrollo del problema

1. Crea un proyecto para la aplicación de reserva
2. Configura la base de datos y las opciones locales (idioma, zona horaria)
3. Sincroniza la base de datos, con el nombre indicado
4. Lanza el servidor interno de django y comprueba que funciona el proyecto
5. Crea la aplicación de reservasAPP
6. Escribe las clases para los modelos
7. Diseña las urls para que se pueda acceder a:
8. Admin
9. Reservas
10. Reservas API
11. Escriba un template genérico que contenga el estilo del sitio para ver los datos. Configura el directorio de templates dentro de settings.py
12. Usa el comando, en el mismo directorio de manage.py, para crear al administrador.
13. Ingresa al menos 5 datos al sistema para cada tabla (Adjunte pantallazos del proceso)

**Aclaraciones:**

* Se deben crear todas las views para cada función solicitada.
* Se deben crear todas las urls necesarias.
* Se deben validar los datos en la API igualmente.