

EXAMEN DEL PARCIAL I
Carrera: Tecnologías de la información

Asignatura: Aplicaciones Web II **Docente:** John Cevallos
Fecha: Noviembre/2021 **Nivel:** Sexto Nivel "A"
Estudiante: ENRIQUE JOSUE CAMPOZANO FIGUEROA **Cédula:** 1311614992 **Firma:**

CALIFICACIÓN EXAMEN: 10 PUNTOS - TIEMPO PARA RESOLVER: 120 MIN

Micro Proyecto de Integración aplicando Node y Express. (10 puntos)

Considere lo siguiente para desarrollar su ejercicio:

- El código debe ser subido en algún repositorio GIT,
- Publicar un video explicativo de los proyectos ejecutados y funcionando (entre 2 y 4 minutos de duración), mostrando su funcionabilidad, y explicando generalidades en el código del mismo.
- Definir los siguientes proyectos: (El número de proyectos puede variar según su análisis)
 - Proyecto que publica sitio web estático.
 - Proyecto Cron Job.
 - Proyecto que publica servicio REST.

Tema:	Matriculación vehicular	%	% Calif.												
Página estática	<ul style="list-style-type: none">Definir un proyecto aplicando Node y Express que publique un sitio web estático que muestre una tabla que contenga por lo menos 10 filas, con la siguiente estructura: <table><tr><th>Placa</th><th>Descripción</th><th>Dueño</th><th>Valor Vehículo</th><th>Impuesto rodaje(1)</th><th>Impuesto verde(2)</th></tr><tr><td>MBB-0788</td><td>Mazda 323 año 2000</td><td>Florinda Mantuano</td><td>5000</td><td>50</td><td>100</td></tr></table>	Placa	Descripción	Dueño	Valor Vehículo	Impuesto rodaje(1)	Impuesto verde(2)	MBB-0788	Mazda 323 año 2000	Florinda Mantuano	5000	50	100	10%	
Placa	Descripción	Dueño	Valor Vehículo	Impuesto rodaje(1)	Impuesto verde(2)										
MBB-0788	Mazda 323 año 2000	Florinda Mantuano	5000	50	100										
Base de datos y modelo con mongoDB	<p>Crear un modelo con Mongoose con la estructura:</p> <ul style="list-style-type: none">✓ Placa✓ Descripción✓ Dueño✓ ValorVehículo✓ ImpuestoRojade✓ ImpuestoVerde✓ Tipo de error <p>Y la funcionabilidad para insertar y consultar las colecciones de MongoDB Atlas, utilizando este modelo.</p>	10%													

Cron Job	<ul style="list-style-type: none"> ✓ Cada 2 minutos captar información de la página web estática utilizando librerías externas (node-cron y axios) y ubicarla en un arreglo listo para ser insertado en cualquier tipo de persistencia. (para facilitar esta tarea puede utilizar cualquier clase o marca sobre el HTML de la tabla). 	20%	
Servicio REST	Definir un servicio REST que presente: <ul style="list-style-type: none"> ✓ Método POST que almacene información utilizando el modelo previamente definido. ✓ Métodos GET que devuelva las colecciones que presenten algún error en el cálculo de los impuestos (Debe corresponder el 1% al rodaje y 2% al impuesto verde los valores correctos). 	20%	
Interoperabilidad e integración	<ul style="list-style-type: none"> ✓ El cron job debe almacenar la información en la base de datos de MongoDB Atlas, utilizando el servicio REST y el modelo en mongoose. ✓ Utilizando Postman conectarse al servicio REST y consultar los movimientos con error. 	30%	
Repositorio GIT	<ul style="list-style-type: none"> ✓ Subir código en repositorio GIT con acceso a joancemac@gmail.com. 	10%	
<u>TOTAL</u>	<u>Suma de todos los porcentajes.</u>	<u>100%</u>	

Forma de entrega:

El URL del proyecto en GIT y el URL de video explicativo del producto final, deberán ser ubicados en un documento formato PDF en el que consten sus datos personales y subidos al aula virtual Moodle (La actividad estará ubicada en la Semana 8).

El video y el repositorio GIT será la evidencia de la entrega en cuanto a su examen.

NOTA: De presentar problemas de conexión al internet se acepta y recomienda trabajar con una versión local de MongoDB, así como llevar GIT en su computadora y tener como evidencia el repositorio local (carpeta oculta. git)

Elaborado por	Revisado y aprobado en área	Recepción Comisión Académica
Ing. John Cevallos Macías Docente	Ing. Edgardo Panchana Flores Colectivo disciplinario Software	Ing. John Cevallos Macías Comisión Académica