

PROGRAMACIÓN WEB

Evaluación N2: Exploración de JavaScript en backend

I. ANTECEDENTES GENERALES

Recinto:	Laboratorio de informática.
Aprendizaje esperado:	Integrar componentes desarrollados en lenguaje JavaScript (JS) con la maqueta de la página WEB pensando en un desarrollo backend.
Criterios de evaluación:	<ul style="list-style-type: none"> - Desarrolla un plan para la integración de los componentes JS en el sistema, asegurando procedimientos y pautas y considerando aspectos críticos como la compatibilidad y el rendimiento en el plan de integración. - Evalúa los componentes JS en relación con la maqueta de la página web, identificando problemas de compatibilidad y proponiendo soluciones. - Integra scripts de JavaScript en una página web, asegurando su funcionamiento en el backend después de la integración. - Comunicación: Comunica sus ideas de manera clara y segura en ámbitos de su quehacer académico y profesional, utilizando vocabulario técnico. - Solidaridad y Fraternidad: Ofrece su disponibilidad y tiempo, con actitud comprensiva hacia los requerimientos de los demás, en distintas situaciones.
Tipo de evidencia / Evidencia:	Proceso: Aplicación de atributos, variables, objetos, condiciones, ciclos y funciones.

II. INSTRUCCIONES Y FORMALIDADES

- **Tipo de evaluación:** Sumativa
- **Tipo de actividad:** Individual
- **Duración:** 80 minutos
- **Ponderación:** 20%
- **Contexto:** Laboratorio de informática
- **Instrumento de evaluación:** Rúbrica

III. DESARROLLO DE LA EVALUACIÓN

Instrucciones Generales

A continuación se te presenta un caso realista que servirá como base para evaluar tu capacidad de análisis, diseño y desarrollo backend utilizando JavaScript y Node.js. Deberás trabajar de forma **individual**, reflexionando a partir del contexto descrito, **identificando requerimientos funcionales y técnicos**, y proponiendo una solución técnica inicial con foco en aspectos de seguridad, modularidad y escalabilidad.

Problema: "Sistema Seguro de Gestión Clínica para la Institución Médica Vitalis"

Contexto General

La **Clínica Médica Vitalis** es una institución de salud privada con más de 20 años de trayectoria, reconocida por su excelencia en la atención primaria y especializada. Actualmente, cuenta con cinco sucursales ubicadas en ciudades estratégicas del país, prestando servicios a más de 50.000 pacientes activos al año. Sin embargo, a pesar de sus altos estándares médicos, su sistema de gestión administrativa y clínica se ha mantenido en una estructura arcaica, lo que ha generado una serie de dificultades operativas.

Entre los principales problemas detectados se encuentran:

- **Duplicación de registros clínicos** por falta de una base de datos centralizada.
- **Retrasos en la programación de citas** y dificultades en el seguimiento de disponibilidad de los médicos.
- **Falta de trazabilidad en los pagos realizados por los pacientes**, lo que ha derivado en conflictos de facturación.
- **Alto riesgo de pérdida de datos** sensibles, debido al almacenamiento en formatos físicos y planillas sin encriptación.
- **Dependencia total del personal administrativo**, lo que impide a los pacientes realizar gestiones por sí mismos.

Como parte de una estrategia de transformación digital impulsada por la nueva gerencia, la clínica ha decidido **desarrollar una aplicación web segura y escalable** que digitalice sus operaciones clave, protegiendo al mismo tiempo la confidencialidad de la información médica de sus pacientes.

Rol del Estudiante / Desarrollador

Has sido contratado como **desarrollador backend** dentro del equipo técnico responsable del proyecto. Tu labor consiste en analizar el caso, extraer requerimientos funcionales y técnicos, y proponer la arquitectura inicial del sistema backend que dará soporte a la aplicación web.

Para ello, deberás aplicar tus conocimientos en desarrollo backend con JavaScript y Node.js, teniendo especial atención a:

- **La seguridad de los datos** (autenticación, sesiones, validación, cifrado)
- **La organización del código backend** (modelo de capas o arquitectura MVC)
- **La definición de rutas, controladores, servicios y estructuras de datos**
- **La correcta identificación de actores y casos de uso**

📌 Parte 1: Producto que Debes Entregar

1. Análisis de requerimientos

- Requerimientos funcionales (qué debe hacer el sistema):
rf1: El sistema deberá de permitir el registro de datos de los pacientes

Rf2: El sistema deberá permitir agendar citas con el personal disponible

rf3: El sistema deberá registrar correctamente los pagos y generar informes.

rf4: El sistema deberá tener un diseño intuitivo y fácil de usar, junto con instrucciones de uso para cada campo, una guía que sirva para que todos los pacientes utilicen el sistema(independientemente de la edad del mismo)

- Requerimientos técnicos (aspectos clave a considerar en la implementación)

Rt1: creación de una base de datos para guardar la información del personal y clientes

Rt2: utilizar la estructura de modelo vista controlador para poder trabajar con varias tecnologías en conjunto y garantizar la creación de un sistema complejo

Rt3: contemplar la creación de una base de datos bien estructurada, para evitar la redundancia y fallos en el sistema que dificulten su gestión y control.

- Identificación de los distintos tipos de usuarios del sistema y sus posibles acciones

Administrador: se encarga de gestionar los datos ingresados y gestionar a todo el personal tomando en cuenta los roles de usuario

Personal: podrán registrarse en el sistema, tendrán un rol importante dentro del sistema y podrán gestionar sus propios datos comunes(horarios, perfil, disponibilidad etc).

Pacientes: podrán registrar sus datos en el sistema y agendar citas con el personal disponible.

2. Propuesta de arquitectura

- Breve explicación escrita del modelo que utilizarás (por ejemplo: modelo de 3 capas o MVC)

Se implementara el backend que contendrá el controlador(controller) que se encargara de conectarse con el modelo(models) el cual hará posible gestionar los datos, como eliminar, agregar, modificar etc y poseera una base de datos con la que se conectara y contendrá los datos que se trabajaran, se integrara la lógica del server y por el lado del frontend se implementara el diseño, y la lógica de comportamiento del mismo en (ejemplo) un archivo html, css y archivos js respectivamente.

- Esquema textual o visual simple (puede ser dibujado) de cómo se organiza el sistema

Backend:

controllers: sistemacontroller.js

Models: sistemaModels.js

Db: base de datos.sqlite

Server.js

Package.JSON

Frontend:

assets/css/js

Css: estilos.css

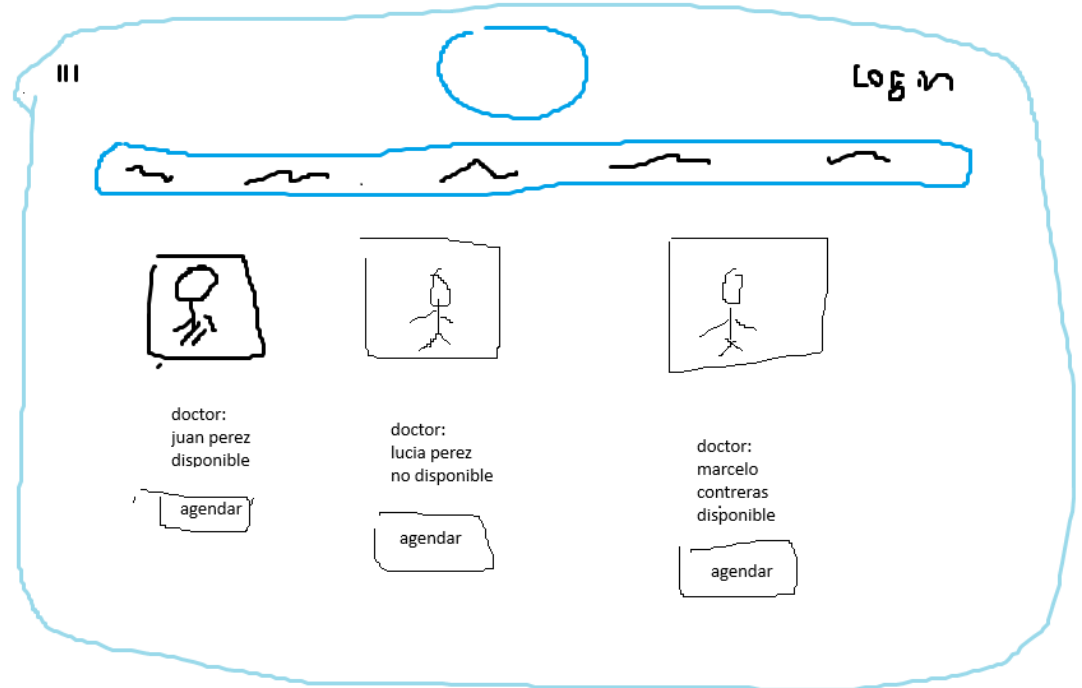
js: script.js
index.html
index2.html
index3.html

3. Propuesta de medidas de seguridad

- Posibles amenazas o vulnerabilidades del sistema
Podría existir la suplantación de datos, o el robo de estos mismos
- Medidas que tomarás para mitigarlas (validación, autenticación, control de acceso, etc.)
Control de acceso para cada usuario, gestión de los mismos con datos personales y únicos solo para validación de registro (verificación de dos pasos o muestra de cara)

4. Simulación de una estructura de datos

- Ejemplo de cómo representarías la información del usuario, cita médica o historial clínico en que formato,



NOTA: IMPORTANTE, todo entregable debe ser realizado por plataforma

Entregable 1: Parte 1: entregada 30 minutos de iniciada la evaluación

Entregable 2: Termino de horario de evaluación: subir avance de proyecto realizado

Entregable 3: Informe y código final 22:00 22-05-2025

De tener un tamaño superior al permitido por plataforma enviar por correo electrónico antes de la hora máxima señalada, además subir en plataforma captura del correo enviado como notificación adicional del medio por el cual fue enviada la actividad