

ANEXO EXAMEN DE CERTIFICACIÓN

Plan Formativo	Desarrollo Aplicaciones FullStack JavaScript V2.0
Objetivo	Desarrollar aplicaciones web Full Stack utilizando tecnología Javascript que den solución a las necesidades de la organización, aplicando las buenas prácticas de la industria para obtener un producto con niveles de calidad requeridos.

Caso “Proyecto Personal”

Contexto del Problema

Al inicio del Bootcamp cada participante, de manera individual, realizó diferentes actividades que le permitieron definir un proyecto o aplicación web a desarrollar, identificando necesidades del usuario final o del mercado, estableciendo objetivos concretos y la viabilidad de realizar el proyecto tecnológico con las competencias técnicas a adquirir a lo largo del proceso formativo.

Objetivo del Proyecto, Alcance y Avance a la fecha

El proyecto busca, dentro de otras cosas, permitir a los participantes demostrar las competencias técnicas necesarias para crear páginas web responsivas utilizando HTML, CSS y JavaScript, siguiendo la arquitectura y codificación de piezas de software en JavaScript que permitan la interacción con el usuario final. La implementación del código deberá seguir los paradigmas de orientación a objetos, orientación a eventos y programación asíncrona. El prototipo del proyecto deberá ser capaz de operar una base de datos relacional utilizando SQL, incorporando el diseño de un modelo relacional de bases de datos. Finalmente, para la implementación del prototipo, se deberá utilizar el entorno multiplataforma Node Express y seguir la lógica de desarrollo de aplicaciones web.

Requisitos Funcionales del Sistema

- El sitio deberá incorporar un menú que permita identificar sus diferentes secciones incluyendo un menú que permita identificar el objetivo u oferta de valor del proyecto.
- Incorporar un sistema o proceso para crear una cuenta en el sitio e iniciar sesión.
- En el caso de que un usuario no esté registrado, el sistema debe redirigirlo a la pantalla para que haga Login o se Registre. Puede usar una pantalla (ruta: /login) con ambos formularios (de registro y de login), o hacerlo en pantallas distintas.
- Deberán existir al menos 2 roles de usuario: Administrador y Usuario, los cuales deberán tener claramente definidos los alcances en la gestión del sitio.(Acceso a mantenedor)
- Almacenar toda la información necesaria en una base de datos SQL.
- Deberá existir una página que permita mostrar un listado de los productos o servicios disponibles (con stock mayor a 0 u ordenados alfabéticamente), además de un NavBar donde pueda realizarse el proceso de registro o iniciar sesión.

- Los productos o servicios deberán contener al menos una foto, el precio (si aplica), la cantidad y la descripción de este.
- Deberá crear una página o mantenedor de productos o servicios, donde se puedan realizar procesos de creación, edición o eliminación, procesos que deben estar condicionados a un rol de administrador del sitios.
- Debes crear una página especial donde cada usuario pueda revisar todos los productos o servicios que ha adquirido.
- Deberá crear una página especial que muestre un listado histórico con el ranking de productos o servicios más demandados.

Definiciones y lineamientos arquitectónicos

Para satisfacer las necesidades del proyecto, se ha definido la utilización de una arquitectura basada en componentes Front-End y Back-end, para lo cual, ha establecen los siguientes lineamientos:

- Para el Backend deberá utilizar Express para disponibilizar la información necesaria.
- El sistema debe incorporar alguna implementación mediante tecnología REST para el consumo de información de APIs.
- Utilizar framework Bootstrap o similar para el manejo de estilos
- El sistema debe utilizar una base de datos PostgreSQL o MySQL.
- La capa de acceso a datos puede hacer uso de un ORM si lo estima conveniente

Requerimientos Funcionales del Sistema

- Las funcionalidades deben estar protegidas para que sean realizadas sólo por usuarios registrados y autenticados correctamente según corresponda.
- Cada producto o servicio deberá contener al menos 1 fotografía.
- Todas las rutas deberán tener una lógica de acceso o de inicio sesión en función del alcance de los roles de usuario establecidos.
- El prototipo de la aplicación debe estar desarrollado en base a convenciones, estilos y buenas prácticas de programación.

Entrega Final del Caso

Una vez implementadas las funcionalidades pedidas, se requiere hacer lo siguiente:

1. **Comprimir proyecto y subir a la plataforma de evaluación.** El archivo comprimido deberá contener sólo código fuente, con su estructura de carpetas ordenada, sin incluir módulos Node y con un archivo .txt que indique el enlace a repositorio de GITHUB.
2. De manera paralela deberá **subir el proyecto a su repositorio GitHub** en base a las siguientes definiciones.
 - i. **Deberá realizar la creación del repositorio:** Inicie creando un repositorio en su cuenta de GitHub. Asigne un nombre descriptivo y relevante al proyecto, y agregue una descripción breve y concisa para comprender su propósito.
 - ii. **Archivo README:** Incluya un archivo README.md en la raíz del repositorio. El archivo README debe contener información sobre el proyecto, como una descripción detallada, instrucciones de instalación, uso, contribución y cualquier otra información relevante.

Además deberá describir e indicar de manera clara y concisa en lugar o piezas de código en donde las implementaciones técnicas dan respuesta a los siguientes criterios.

a) Sitio web (HTML, CSS y JS)

- Utilización de tags html, estilos y responsividad
- Utilización de Bootstrap

b) Algoritmia de cálculo y manipulación de archivos de texto:

- Utilización general del lenguaje, sintaxis, selección de tipos de datos, sentencias lógicas, expresiones, operaciones, comparaciones.
- Utilización de sentencias repetitivas.
- Utilización de estructuras de datos.
- Manipulación de archivos.

c) **Consultas a la base de datos de la aplicación:**

- Manipulación de columnas requeridas desde diferentes tablas para presentar la información solicitada.
- Utilización de JOIN para relacionar la información de distintas tablas.
- Utilización de WHERE para filtrar la información requerida por la aplicación.
- Utilización de cláusulas de ordenamiento para presentar la información.
- Utilización de cláusulas de agrupación de información para obtener datos agregados.

d) **Lenguaje Node**

- Inclusión de paquetes y librerías de usuario.
- Lectura de parámetros de entrada
- Agrupación del código y separación por funcionalidad.
- Utilización de funciones asíncronas
- Funcionamiento general adecuado del aplicativo

e) **Persistencia y conexión a Base de Datos**

- Manejo de conexión a base de datos desde Node.
- Manejo y ejecución de consultas desde Node.

f) **Uso de Express**

- Creación de un servicio Rest con Express

iii. **Estructura de archivos y carpetas:** Organice el proyecto en una estructura de archivos y carpetas clara y lógica. Asegúrese de que todos los archivos estén en las ubicaciones apropiadas y sigan las convenciones de nombres establecidas.

3. **Documentación adicional:** Opcionalmente, proporcione documentación adicional si es necesario, tales como guías de usuario u otra documentación técnica.