

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
(ESPOL)**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y
COMPUTACIÓN**

PROYECTO FINAL

**DESARROLLO DE APLICACIONES WEB
(CCPG1010)**

**TÉRMINO ACADÉMICO
2019 - II**

Contenido

Objetivos de aprendizaje del curso.....	3
Objetivo de la actividad.....	3
Lineamientos	3
Validación de usuarios.....	4
Plazos.....	4

Objetivos de aprendizaje del curso

- Implementar una aplicación web que utiliza estándares web para presentación (ej. CSS) y para programación cliente (ej. librerías JavaScript).
- Diseñar aplicaciones web que manejan requerimientos asincrónicos (ej. AJAX) para actualizar la interfaz sin refrescar la ventana del navegador.
- Implementar una aplicación web que utiliza el patrón modelo vista controlador
- Explicar el concepto de sesiones web y canales de comunicación seguros en la implementación de aplicación que manejan estados en el servidor.

Objetivo de la actividad

Diseñar e implementar una aplicación Web bajos los estándares y consideraciones de diseño revisadas durante el segundo parcial mediante el uso de metodologías modernas en el desarrollo.

Lineamientos

- **Básico**
 - **Usuarios.** Su aplicación web considerará, al menos, dos tipos de usuarios. Ustedes definen los atributos por cada tipo de usuario.
 - **Administrador.** Es el que se encargará de subir contenidos del lado del servidor. Existe un usuario por este tipo.
 - **Cliente.** Tendrá acceso a una zona específica donde podrá ver su perfil (con 5 características) y, mínimo 2 páginas de acceso exclusivo relacionado a la aplicación web. Existirán al menos 4 usuarios por este tipo.
 - **CRUD.** El tipo de usuario Cliente tendrá una página/sección que permita crear, recuperar, actualizar y eliminar un recurso relacionado con la aplicación web.
 - **API.** Los requerimientos se deben realizar mediante un API, de acuerdo con el método HTTP correspondiente. Al menos 4 entidades deben accederse mediante el API, cada uno por GET, POST, UPDATE y DELETE. No se considerarán como parte de respuesta a este requerimiento a los módulos de administración que vienen en el web framework.
- **Estadísticas.** Al menos, tres gráficos estadísticos con D3js, en el lado del cliente, cuya información debe ser consumida desde un API en el lado del servidor.
- **Envío de datos por correo.** Debe proveer de un formulario; por ejemplo, un formulario de contáctenos, que permita el envío de información desde una página en el cliente Web al correo del administrador.

- **Administración.** Utilice los módulos/paquetes por defecto que le ofrece su *web framework* para la manejar la administración de su aplicación web. Además de la autenticación y autorización.
- **Framework en el cliente.** Su proyecto **debe** utilizar un *framework* en el lado del cliente; por ejemplo: Angular2, Emberjs, Reactjs, Backbonejs, etc.
- **Reportes.** Su aplicación deberá generar un reporte cuyos resultados deben vincular los datos de una Base de datos relacional con una Base de datos no relacional. Para esto, desde el *frontend* se mostrarán los controles (calendarios, combobox, dropdown list, etc) necesarios para generar los reportes.
- **Reportes con formato (Mejoramiento).** Además, será posible descargar los reportes en formato PDF y/o XLSX.
- **Tiempo real (Mejoramiento).** Defina una sección que maneje un recurso en tiempo real y que los usuarios puedan acceder al mismo, para realizar algún cambio; por ejemplo, la aplicación de restaurantes tendrá un servicio de pedidos. Los clientes podrán ver que se disminuyen el número de platos de los cuales pueden pedir. Utilizará WebSockets.

Validación de usuarios

El sitio web debe ser validado por 5 usuarios reales, de acuerdo a los formatos provistos en clase. Además de las evidencias del caso.

Plazos

Realiza un plan de plazos semanales para revisiones que tendremos de cada viernes; además, considera la fecha de entrega del proyecto.