

PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA
FACULTAD DE CIENCIAS E INGENIERÍA
ESCUELA DE INGENIERÍA EN COMPUTACIÓN Y TELECOMUNICACIONES

Programación Web - ICC-352
Proyecto Final

Objetivos:

1. Aplicar todos los conocimientos básicos y fundamentales sobre HTML, CSS, Javalin y JavaScript.
2. Abstraer los requerimientos básicos de un cliente y realizar el modelo, diseño e implementación del proyecto .

Desarrollo.

Los estudiantes realizarán cada uno de los requerimientos asignados a sus respectivos proyectos. Deben cuidar la presentación del sistema así como la funcionalidad, siempre dándole más prioridad a este último. Pueden utilizar una plantilla CSS para tener un producto más acabado. La gestión de los cambios deben ser trabajados en el control de versiones sobre **GIT**. Los estudiantes deben enviar sus cambios sobre el mismo proyecto que están implementando y trabajar de una manera colaborativa.

Forma de Entrega:

La entrega será una presentación del proyecto en (10 minutos) con un vídeo bajo la plataforma youtube con **su respectivo reporte**; la presentación debe cubrir cada unos de los requerimientos solicitado en el proyecto trabajado. La base de datos a utilizar será el motor H2 en modo servidor. Deben enviar el proyecto al repositorio del servidor de control de versiones y tener disponibles el script para la generación de la base de datos de forma automática la primera vez que se ejecuta conjunto **al diseño entidad-relación**.

Evaluación de los proyectos:

Los proyecto realizados deben utilizar las siguientes tecnologías como mínimo:

- HTML
- CSS
- Plantilla
- Modelado de objetos
- Javalin
- ORM
- JavaScript
- Publicación de proyecto

Entre mas innovación e investigación utilicen en el proyecto, podrán ganar puntos extras. Tienen libertad para agregar requerimientos que no se encuentren en los planteados.

Sistema para Acortar URL

El sistema a desarrollar será utilizado para obtener una URL y devolver una nueva URL desde el dominio del proyecto donde se muestre los accesos, referencia, gráficos de uso, etc. Ver la funcionalidad del proyecto Bitly <https://bitly.com/>.

Los requerimientos que debe contar la aplicación son:

1. El sistema debe permitir que los visitantes al sistema puedan registrarse con la finalidad de tener acceso las funcionalidades del sistema donde sea requerido. De manera automática, se creará un usuario administrador y éste tendrá la potestad de convertir un usuario registrado con sus mismo privilegios. El usuario administrador no puede ser eliminado.
2. Los usuarios tanto registrados como no, podrán utilizar la funcionalidad de poder a cortar una URL, indicando la URL original y el sistema generando la nueva, llevando un control de la cantidad de veces que es accedido, los navegadores por los cuales son visitados, la IP o el dominio del cliente y plataforma de sistema operativo accedido.
3. Solo los usuarios registrados pueden tener acceso en todo momento a la cantidad de posteos realizados y ver la información resumen del enlace que fue registrado. Los usuarios no registrados solo tendrá disponibles esta funcionalidad hasta que su sesión este disponible, pero los enlaces generados deben estar disponibles.
4. En la pagina de resumen de la información debe mostrar un gráfico con la hora y fecha a la cual fue accedido el enlace. Puede utilizar librerías en Java para estos fines del lado del servidor o utilizar el API o librerías que proporcionan esa funcionalidad, pueden ver [Google Chart](#) o [Chartjs](#). Son libres de utilizar la librería que consideren.
5. En la pagina de resumen de la información debe ser mostrado el gráfico mediante el código QR ([Wikipedia](#)), esta integración puede ser realizada mediante las diferentes librerías que cuenta el lenguaje del lado del servidor ([Ver Post](#)) o desde el cliente como son: [Qrcodejs](#) o [VanillaQR.js](#). Tiene que estar disponible el enlace para acceso directo.
6. Solo los usuarios administradores tiene acceso a poder eliminar los enlaces posteados si son considerados ofensivos.
7. A la hora de agregar un nuevo enlace, el sistema debe presentar una vista previa del recurso. Pueden ver las siguientes librerías [Microlinkhq](#) o [Link preview](#), entre otros.

8. Crear un servicio REST, SOAP y gRPC que realice las siguientes operaciones:
 - (a) Listado de las URL publicadas por un usuario incluyendo las estadísticas asociadas.
 - (b) Creación de registro de URL para un usuario retornando la estructura básica (url completa, url acortada, fecha creación, objeto de estadística y la imagen actual del site (vista previa) en base64.
9. Para el servicio REST es necesario implementar un esquema de seguridad basado en JWT.
10. Crear un cliente en cualquier plataforma que implemente las operaciones publicadas por el API REST.
11. Crear un cliente en cualquier plataforma que implemente las operaciones publicadas por el API SOAP.
12. Crear un cliente en cualquier plataforma que implemente las operaciones publicadas por el API gRPC.