
Universidad Tecnológica Metropolitana.

Departamento de Computación e Informática.

Computación Paralela y Distribuida

Profesor: Sebastián Salazar Molina.

Proyecto REST

07 de diciembre del 2022

VISIÓN GENERAL

En el mundo tecnológico en el que nos desenvolvemos, las aplicaciones móviles tienen cada vez más importancia en el mercado, por lo que el desarrollo de servicios backend (que apoyan este tipo de aplicaciones) es de vital importancia actualmente, por lo que los servicios rest han adquirido un lugar preponderante en la estrategia de este tipo de soluciones. Estas aplicaciones pueden estar pensadas para presentar información directamente a los usuarios finales, para interoperar con otros sistemas o una mezcla de estos comportamientos.

OBJETIVOS

1. Comprender el funcionamiento del protocolo HTTP (sus verbos y estados).
2. Comprender el funcionamiento de aplicaciones stateless, mecanismos asíncronos y funcionamiento REST.
3. Diseñar e implementar un proyecto de software con requerimientos específicos.

ESPECIFICACIONES

En el escenario supuesto de que la Universidad Tecnológica Metropolitana, está evaluando una forma de implementar un sistema de democracia digital (para consultas, paros, encuestas, etc). Para esto se le solicita construir un api rest que permita realizar esta tarea. El sistema debe contemplar lo siguiente.

Usuarios.

El sistema debe contener los usuarios de la UTEM, se necesita que el mecanismo de autenticación sea el provisto por Google (oauth 2.0), lo que garantiza la validez de la autenticación, es necesario además que exista un manejo de roles, para una personalización futura. Cada llamada a los servicios api, debe identificar al usuario y tener un access token, para validar la llamada contra los servicios de Google, esto se debe realizar usando una cabecera jwt.

Consultas.

Se necesita registrar las consultas, los elementos mínimos necesarios son disponer de un token único que permita identificar esta consulta para la interacción de los sistemas y un nombre para identificar la consulta de forma humana y un flag que permita determinar si dicha consulta está activa o no. Por otro lado, cada opción asociada a la consulta debe tener un atributo numérico (que se usará como identificador de la opción) y un campo para el texto que describe la opción.

En esta etapa se permite que la carga de las consultas y sus detalles, se realicen directamente a la capa de persistencia de datos (Por ejemplo, se permita la carga por un script sql).

Votos.

Se debe poder registrar el voto, es necesario controlar que sólo se pueda realizar un voto por usuario y consulta, por otro lado el diseño debe garantizar el anonimato de la elección.

API

El api a construir debe tener algunas consideraciones generales para su implementación.

- El mecanismo de transferencia debe ser Json.
- Cualquier error (controlado o no) debe producir salidas en formato Json.
- Las operaciones deben estar autenticadas.
- La versión de referencia se encuentra en:
<https://api.sebastian.cl/vote/swagger-ui/index.html>
- El swagger se encuentra en: <https://api.sebastian.cl/vote/v3/api-docs>

Autenticación.

Se necesita una operación que permita la autenticación contra google y almacenar los datos necesarios para validar el token de acceso provisto por Google. Se recomienda que las operaciones de entrada, aquellas que interactúan con Google, están protegidas al menos por credencial/contraseña. La validación se realizará con un objeto jwt¹ que se incluirá en una cabecera “authorization”.

Votación.

Las operaciones mínimas necesarias, son:

/voter/polls

Se requiere lo siguiente:

- La operación debe ser **GET**.
- Cabecera de autorización
 - Authorization: Bearer jwt.
 - Se debe validar en base a este token (provisto por Google), el usuario y si su token está activo o no.
- La salida debe indicar aquellas encuestas con al menos la siguiente información:
 - **token**. Identificador de encuesta.
 - **nombre**. El nombre de la encuesta.
 - **activo**. Flag que indica si la encuesta está vigente o no.
 - **opciones**. Listado de opciones asociadas a la encuesta.

¹ <https://developers.google.com/identity/sign-in/web/backend-auth>

-
- número de la opción.
 - descripción de la opción.

/voter/vote

Esta opción debe permitir registrar un voto, es importante que esto sea anónimo.

- La operación debe ser **POST**.
- Cabecera de autorización
 - Authorization: Bearer jwt.
 - Se debe validar en base a este token (provisto por Google), el usuario y si su token está activo o no.
- El cuerpo de la petición debe contener:
 - Token de la encuesta.
 - El número de la selección.
- La salida debe responder con código 200 ó 201 y un objeto que indica que la operación fue exitosa.

/voter/{token}/results

Esta opción debe permitir desplegar los resultados del proceso.

- La operación debe ser **GET**.
- Cabecera de autorización
 - Authorization: Bearer jwt.
 - Se debe validar en base a este token (provisto por Google), el usuario y si su token está activo o no.
- Cómo parte de la url se debe indicar el uso del token de la encuesta.
- La salida debe responder con un código 200 e indicar:
 - El nombre de la encuesta.
 - Un listado con el nombre de la opción y la cantidad de votos acumulados.

Equipo.

El proyecto deberá ser realizado grupal, con los equipos conformados en clases.

Código.

El código debe ser entregado a más tardar, el 07/12/2022 hasta las 23:59:59.999 horas de Santiago de Chile, este proyecto debe estar publicado en **repositorio grupal de github** (privado con invitación al docente).

Informe.

Se debe presentar un informe completo que explique perfectamente el diseño de la solución, con la estructura mínima esperable es:

- Debe tener una introducción.
- Debe indicar la forma de resolver el problema. En particular, se debe detallar el mecanismo que garantice el anonimato, debe incluir imágenes que expliquen la arquitectura, diagramas de secuencia, diagramas de Entidad Relación, diccionario de datos y en definitiva todo lo necesario para explicar la solución entregada.
- Debe indicar y justificar la tecnología usada.
- Debe tener una conclusión clara y acorde a lo establecido en el desarrollo del informe.
- **El informe corresponderá a una nota independiente y deberá ser enviado al docente a más tardar el Lunes 05 de diciembre de 2022 hasta las 23:59:59.999 hora continental.**

EVALUACIÓN

Documentación.

Parte de la evaluación consiste en la documentación de las funciones. Que debe ser clara, concisa y descriptiva de lo que el código realiza. El formato puede ser swagger o openapi.

Código

El código debe ser claro, fácil de leer, ordenado y cumplir con buenas prácticas de programación, se inspeccionará el código fuente.

Resultados.

Un criterio de evaluación que se tomará en consideración: el tiempo de ejecución de la tarea y el cumplimiento de los objetivos planteados. En aquellos puntos con medida cuantitativa menos es mejor. La evaluación es porcentual.

Implementación.

El proyecto es grupal, debe realizarse por los distintos equipos formados en la asignatura.

Cada grupo puede implementar la aplicación en el lenguaje que prefiera, siempre y cuando sea un lenguaje open source, y que se pueda instalar nativamente (desde el repositorio base) en Ubuntu 22.04 LTS de 64 bits. El proyecto debe cumplir con todas las normativas de licenciamiento aplicables.