LABORATORIO 7

ESCUELA COLOMBIANA DE INGENIERÍA

CICLOS DE VIDA DEL DESARROLLO DE SOFTWARE - CVDS

Finalización CI/CD - Manejo de Data - ORM

PARTE I. FINALIZACIÓN CI/CD

 # Verificación APP

- Verificar funcionamiento de app MyShutle en mi cuenta de la nube de Azure.
- Al presionar en su URL: ej.´https://mycvdsappname.azurewebsites.net/myshuttledev´
- Deberá salir un pantalla mostrando la pantalla de login de MyShuttle. En caso que no, qué es lo más probable, vamos a revisar cómo mirar los errores.

Troubleshooting - Activando logs

- Revisión de logs del sistema
  - Ingresar al los recursos de nuestra cuenta en la nube de Azure.
  - Entrar al recurso de nuestra aplicación web.
  - Buscar la opción de Logs Stream, ingresar allí. Nos debe salir un mensaje que nos indica que primero debemos activar ó configurar los logs.
  - Volver a la página anterior y presionar en la opción Configure Logs.
  - Vamos a configurarlos de la siguiente manera:
    - Application logging: ON (Level: Error)
    - Application login (Blob): OF
    - Web server logging: File System
    - Quota 35 MB
    - Retention Period: 3
    - Detailed error messages: ON
    - Failed request tracing: ON
    - —> Guardar.
  - Revisar qué está pasando a la hora de hacer login.
- Revisión del código fuente
  - Descargue el repositorio de MyShuttle de su cuenta de AzureDevOps (ADO).
    - Busque el repositorio MyShuttle en su cuenta de ADO
    - Ejecute el comando git clone para descargar el proyecto.
    - Sí le solicita una contraseña, presione el botón Generate Git Credentials y obtenga la contraseña generada para cuando se la solicite localmente Git.

Continuous Integration - Activando la opción integración continua en el Pipeline.

- Buscar en ADO los pipelines, y editar el pipeline de nuestro proyecto.
- Activar el checkbox de “continuous integration” en el menú Triggers. —> Guardar.
- Ahora cuando hagamos un push, se ejecutará el pipeline de construcción de la nueva versión de nuestro proyecto.

- Para desplegarlo, ir al Menu Releases y manualmente crear un release.
- Luego de que sea exitoso podemos volver a probar nuestro sistema.

Bugfixing - Solucionando un Bug
- Revisar en qué parte está ubicada la información de la base de datos.
- Revisa las propiedades de conexión del proyecto java a la base de datos.
- Validar cuales son los datos de conexión correctos.
- Actualizar información de conexión a la BD en la clase.
- Subir los cambios al repositorio remoto.
- Validar si ya funciona el ajuste.

PARTE II. USANDO SPRING DATA DESDE CERO

- Revisemos las clases: DataAccess.java y veamos la manera en la que se crearon los métodos: login, employeeFares, getFareTotal.
- Actualmente el proyecto se conecta a la base de datos usando JDBC (Java Data Base Connection)
- Crea un nuevo repositorio en tu cuenta de github llamado cvds-7, y sigue las instrucciones del siguiente tutorial de spring.
  - https://medium.com/@saultobias13/a-quick-start-with-spring-boot-and-spring-data-jpa-32718a8f4706
- Crearemos una base de datos local usando Docker:
  - Descargar imagen de MySQL
    - 'docker pull mysql'
  - Correr contenedor de MySQL
    - 'docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:tag
  - Descargar un cliente de base de datos: DBeaver
    - Crear una tabla de la base de datos: EMPLOYEE.
      - Con las siguientes columnas:
        - EMPLOYEE_ID (VARCHAR)
        - FIRST_NAME (VARCHAR)
        - LAST_NAME (VARCHAR)
        - ROLE (VARCHAR)
        - SALARY (DOUBLE)
- Buscar cómo conectar nuestro proyecto de spring a una base de datos MySql.

ENTREGA

- CI/CD: La URL pública de su cuenta en la nube de Azure que apunte a la aplicación de MyShuttle. Se solicitará su activación para revisión.
- DATA: URL del repositorio en github funcionando con una base de datos local.