

## Guía de Criterios de Revisión

PROYECTO: **UrQuery Etapa I** CODIGO-GRUPO: \_\_\_\_-\_\_\_\_ FECHA:24/10/2022

### MIEMBROS GRUPO QUE TRABAJARON

- 1) NOMBRE: \_\_\_\_\_ ID: \_\_\_\_\_ Presente: \_\_\_\_\_  
2) NOMBRE: \_\_\_\_\_ ID: \_\_\_\_\_ Presente: \_\_\_\_\_  
3) NOMBRE: \_\_\_\_\_ ID: \_\_\_\_\_ Presente: \_\_\_\_\_  
4) NOMBRE: \_\_\_\_\_ ID: \_\_\_\_\_ Presente: \_\_\_\_\_  
5) NOMBRE: \_\_\_\_\_ ID: \_\_\_\_\_ Presente: \_\_\_\_\_

---

### CRITERIOS REVISION Y VALORACIÓN: Nota Total: \_\_\_\_\_

#### (1) CUMPLIMIENTO (Cumplen o no cumplen)

- a) Presentaron esta hoja en PDF llena en lo que les corresponde: \_\_\_\_\_  
b) Demo sin retrasos:(0(15) min) \_\_\_\_\_  
c) Se envió entregable cuando y según fue pedido: \_\_\_\_\_

**NOTA:** Incumplimiento de cualquiera a)-c) anula todo el proyecto y reciben un cero. Ni siquiera se revisa el resto. E

---

#### (2) REVISION DE RESULTADOS HACIENDO DEMO (máximo 60pts) Puntos Obtenidos: \_\_\_\_\_

**Tarea 1):** Revisión empieza usando el mismo zip subido al DRIVE por el coordinador. Si no es el mismo ahí termina la demo, nota cero en el proyecto

**Tarea 2):** Se descomprime el zip en alguna carpeta. Se abre y muestra un README.md y se siguen las instrucciones ahí indicadas para compilar y levantar la aplicación.

**Tarea 3):** Se levantan correctamente los servidores según el README. Son dos: el de java y el de Prolog.

**Tarea 4):** Se carga la página ruta / o /index.html. La página tiene el gesto para un about y las áreas DA, EA y RA y los controles para pedir un documento (DA) y para compilar un script (EA). DA y EA permiten editar texto.

**Tarea 5):** Se chequea gesto de about que muestra los datos de los autores de una manera presentable y apropiada. Se muestra que sí se hizo el request (por GET) usando el developers tool al invocar el gesto.

**Tarea 6):** Se prueba que si se usan los controles de enviar o compilar y no hay datos la página asociados se reportan los errores amigablemente (no alert). Puede ser un diálogo o un status bar o similares.

**Tarea 7):** Se prueba pedir un documento y el server de java responde con un fake que aparece en DA. Request GET. Puede estar alambrado en código o tomarse del Filesystem.

**Tarea 8)** Se edita algo en EA. Se envía a compilar lo que sea que esté en EA que vuelve del server con un timestap (eco). Se demuestra que eso provoca un request fake a Prolog desde Java. Se prueban que se hacen los request en el browser usando Developers tools. El request de compilar es por POST.

**Tarea 9)** Se repiten las tareas anteriores para probar que todo funciona con distintos envíos.

**Tarea 10)** Se demuestran posibles extras, si hay alguno. Sólo se aceptan, si lo anterior 1-9 funciona en un alto porcentaje.

---

**EVALUACIÓN FUNCIONALIDAD POR EL DOCENTE DURANTE DEMO:**

(T1) \_\_\_\_\_

(T2) \_\_\_\_\_

(T3) \_\_\_\_\_

(T4) \_\_\_\_\_

(T5) \_\_\_\_\_

(T6) \_\_\_\_\_

(T7) \_\_\_\_\_

(T8) \_\_\_\_\_

(T9) \_\_\_\_\_

(T10) \_\_\_\_\_

---

**(3) Revisión offline de entregable (máximo 40 pts.) Puntos Obtenidos:** \_\_\_\_\_

**Criterios:** Se refiere a revisar el código propio entregado. "Calidad" del código, estructura de proyecto (README), limpieza del HTML, apego a paradigma FP-OOP, estilo; modularidad; código DRY y Knuth; combinadores propios, comentarios; eficiencia, otros. **Reciben cero si hay código**

**que no tiene nombre o no se muestra una estructura de proyecto ordenada que permita la revisión decentemente.**

**EVALUACIÓN SUBJETIVA DEL DOCENTE:**

- (1) \_\_\_\_\_
  - (2) \_\_\_\_\_
  - (3) \_\_\_\_\_
  - (4) \_\_\_\_\_
  - (5) \_\_\_\_\_
  - (6) \_\_\_\_\_
  - (7) \_\_\_\_\_
- 

**(4) Extras (hasta 20pts) Sólo si cumplen con un 90% en lo obligatorio. Debieron ser aceptados como válidos por el docente primero. Si son 5 debe haber un extra que no suma.**

**EVALUACIÓN SUBJETIVA DEL DOCENTE:**

- (1) \_\_\_\_\_
- (2) \_\_\_\_\_
- (3) \_\_\_\_\_
- (4) \_\_\_\_\_
- (5) \_\_\_\_\_