

## Guía de Criterios de Revisión

PROYECTO: **ENano Parte II** CODIGO-GRUPO: 06-10 FECHA REV\_PREVISTA:17/11/2020.

### MIEMBROS GRUPO

- 1) NOMBRE: Braslyn Rodríguez Ramírez ID: 402420750 Presente: \_\_\_\_\_
- 2) NOMBRE: Philippe Gairaud Quesada ID: 117290193 Presente: \_\_\_\_\_
- 3) NOMBRE: Enrique Méndez Cabezas ID: 117390080 Presente: \_\_\_\_\_

---

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

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

- a) Presentaron esta hoja PREVIAMENTE llena en lo que les corresponde: \_\_\_\_\_
- b) Demo sin retrasos:(0(10) min)\_\_\_\_\_
- c) Se envió entregable cuando y según fue pedido: \_\_\_\_\_
- d) Se entregó reporte escrito BAT2020-Gradle-Junit-Mocha según fue pedido:\_\_\_\_\_

**NOTA:** Incumplimiento de cualquiera a)-c) **anula todo el proyecto** y reciben un cero. Ni siquiera se revisa el resto. El reporte tendrá su revisión independiente.

---

#### (2) REVISION DE RESULTADOS DEMO (60pts. El puntaje de cada tarea dependerá del rendimiento promedio de todos los grupos, siendo 4, 5 y 6 las más importantes y prioritarias y 1 y 2 necesarias para recibir nota del todo.

**Tarea 1):** Revisión empieza usando el mismo zip entregado en DROPBOX. Se prueba que tiene el mismo CRC-SHA del que tiene el profesor. Si no coincide ahí termina la demo, nota cero en el proyecto

**Tarea 2):** Siguiendo el README, se construye el proyecto con Gradle (y .bats). Si el build es fallido ahí termina la demo, la nota es cero en el proyecto. Igualmente es cero si no hay README o este no es preciso.

**Tarea 3):** Se levantan (de manera simple) los tres servidores (estático, servicios Java y Prolog) y la página SPA Todo lo anterior que compete (Parte-I) funciona, y ahora hay área de evaluación

**Tarea 4):** Se teclea un archivo Enano (digamos por default `File.no`) válido. Se envía al servicio de transpilación en `Prolog`. Este transpila digamos a `File.java` y lo envía al servicio de compilación `Java` que le retorna sus resultados. Estos aparecen en el área de output de la página,

es decir, el `File.java` y la salida del compilador de Java. Un gesto de página, permite cambiar el nombre default que recibe el archivo `.no`.

**Tarea 5):** En el área de evaluación se puede probar que el código asociado a `File.no` ejecuta. Esto se hace tecleando `File.main()`. El resultado aparece en el área de evaluación al estilo de un SHELL. El área de evaluación permite clear.

**Tarea 6):** Se teclea un `File.no` (nombre hipotético) que tiene errores. Se prueba el análogo del paso previo en este caso, donde no hay compilación `Java`. Los mensajes de error son los emitidos por el transpilador mostrados en el output de la página. Si se trata de evaluar, se indica que no es posible. Note que si los errores son de transpilación estos deberían ser detectados por el compilador de Java y deberían aparecer en output también.

**Tarea 7)** Se repiten las tareas anteriores para probar que todo funciona después de varios casos de distinta forma y complejidad.

**Tarea 9):** En todo momento se puede verificar que el pipeline de tres servidores independientes está funcionando como se espera. El de Java usa el `JavaCCAPI`. Cualquier error de servicio es manejado de manera controlada que se pueda visualizar en la página si ahí corresponde y consolas de servidores si es de backend. No hay errores de CORS.

**Tarea 10).** Si se han transpilado varios archivos durante la sesión, con distintos nombres, es posible evaluarlos por su nombre. Si se repite un nombre, le cae encima al anterior, se espera con advertencia al usuario.

---

#### EVALUACIÓN FUNCIONALIDAD POR EL DOCENTE:

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

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

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

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

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

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

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

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

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

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

---

**(3) Revisión offline de entregable y de participación grupal(40 pts)**

**Criterios:** Se refiere a revisar el código propio entregado. "Calidad" del código: apego a un build Gradle; apego a paradigma (FP y LP), estilo; modularidad (JS y Prolog); código DRY y Knuth; comentarios; eficiencia, otros. **Reciben cero si hay código que no tiene nombre o no es un proyecto Gradle o no se puede construir o ejecutar por razones ajenas al evaluador.**

**NOTA:** Se podrán hacer preguntas sobre el proyecto a miembros específicos cuya respuesta INDIVIDUAL podrá considerarse coma si fuera grupal. Dichas respuestas pueden determinar fuertemente este rubro o incluso el del proyecto completo.

**EVALUACIÓN SUBJETIVA DEL DOCENTE:**

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

---

(4) Extras (hasta 20pts) Sólo si cumplen con un 90% en lo obligatorio. Debieron ser autorizados antes de entregar por el docente primero. Si son 5 debe haber un extra que no suma. Se deben anotar acá por cada grupo, sino se anotan no se toman en cuenta.

**EVALUACIÓN SUBJETIVA DEL DOCENTE:**

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