

# Plan de gestión de calidad:

*“Bags, please!”*

## **Hard Break devs**

Julia Cajigal Mimbrera

Fernando Moreno Díaz

Denis Gudiña Núñez

Luis Miguel Moreno López

## Índice

---

Línea base de calidad del proyecto (métricas de calidad) .....	3
Plan de mejoras del proyecto .....	4
Lista de verificación de calidad .....	5
Roles para la gestión de la Calidad .....	8

## LÍNEA BASE DE CALIDAD DEL PROYECTO (MÉTRICAS DE CALIDAD)

En este plan de gestión de calidad pretendemos establecer cómo se implementarán las políticas de calidad establecidas para el proyecto. Aquí también indicamos los valores de tolerancia respecto a cada apartado de cara al producto final y las medidas que usaremos para cuantificar cada factor.

Factor de calidad	Objetivo de Calidad	Métrica	Frecuencia y momento de medición	Frecuencia y momento de reporte
Errores en el código	EC $\geq$ 0.98	EC = errores código / errores permitidos	<ul style="list-style-type: none"> <li>● Frecuencia semanal.</li> <li>● Medición miércoles.</li> </ul>	<ul style="list-style-type: none"> <li>● Frecuencia semanal.</li> <li>● Medición miércoles.</li> </ul>
Cumplimiento cronograma	CC $\geq$ 0.95	CC = cumplimiento actual cronograma / cumplimiento esperado	<ul style="list-style-type: none"> <li>● Frecuencia semanal.</li> <li>● Medición miércoles.</li> </ul>	<ul style="list-style-type: none"> <li>● Frecuencia semanal.</li> <li>● Medición miércoles.</li> </ul>
Eficiencia	E $\geq$ 0.98	E = segundos tiempo ejecución aplicación /	<ul style="list-style-type: none"> <li>● Frecuencia semanal.</li> <li>● Medición miércoles.</li> </ul>	<ul style="list-style-type: none"> <li>● Frecuencia semanal.</li> <li>● Medición miércoles.</li> </ul>
Cumplimiento costo	CO $\geq$ 0.95	CO = costo actual proyecto / costo estimado	<ul style="list-style-type: none"> <li>● Frecuencia semanal.</li> <li>● Medición miércoles.</li> </ul>	<ul style="list-style-type: none"> <li>● Frecuencia semanal.</li> <li>● Medición miércoles.</li> </ul>
Testing juego (bugs)	B $\geq$ 0.98	B = cantidad de bugs encontrados actualmente / cantidad de bugs permitidos	<ul style="list-style-type: none"> <li>● A partir de etapa de testing.</li> <li>● Periódicamente, cada 1-2 días.</li> </ul>	<ul style="list-style-type: none"> <li>● A partir de etapa de testing.</li> <li>● Periódicamente, cada 3-4 días.</li> </ul>
Testing aplicación	A $\geq$ 0.95	A = número de pruebas satisfactorias realizadas /	Final del proyecto	Final del proyecto
Accesibilidad y usabilidad	AU $\geq$ 0.95	AU $\geq$ número de pruebas superadas a través de validadores con $>$ 0.95 / número de pruebas superadas a través de validadores con $<$ 0.95	Final del proyecto	Final del proyecto

## PLAN DE MEJORAS DEL PROYECTO

---

El plan de mejora de procesos pretende especificar los pasos a seguir para identificar actividades que son susceptibles de mejora o que causan desperdicios en el proceso de desarrollo.

1. **Delimitar el proceso:** determinaremos y acotaremos el proceso que es objeto de evaluación y por tanto de posible mejora.
2. **Determinar la oportunidad de mejora:** para cada proceso evaluaremos la medida en que este puede mejorarse. Una posibilidad es que la reestructuración no merezca la pena en porcentajes de mejora mínimos apenas perceptibles en cuyo caso no seguiremos profundizando en el desempeño y características de dicho proceso.
3. **Tomar datos sobre el proceso:** se evaluará con detenimiento el proceso, así como los puntos débiles y fuertes de este.
4. **Definir acciones correctivas:** una vez detectados los problemas plantearemos una serie de medidas enfocadas en suplir carencias y mejorar el proceso.
5. **Aplicar medidas correctivas:** se aplicarán las medidas planteadas.
6. **Comprobación resultados:** una vez aplicadas las medidas comprobaremos si efectivamente estas han supuesto una mejora del proceso.
7. **Estandarizar resultados:** si hemos logrado nuestro objetivo estandarizaremos las mejoras para hacerlas parte de nuestro proceso.

## LISTA DE VERIFICACIÓN DE CALIDAD

En este apartado especificamos para cada paquete de trabajo si existe un estándar asociado. También establecemos las actividades de control y revisión así como los requisitos con los que debe contar cada entregable.

Código actividad	Actividad	Estándar de calidad aplicable	Actividades de prevención	Actividades de corrección	Requisitos
1.1	Recopilación ideas		Revisión estándar	Revisión/aprobación por los miembros del proyecto	Según establecimiento del equipo.
1.2	Reuniones		Revisión estándar	Revisión/aprobación por los miembros del proyecto	Según establecimiento del equipo.
2.1	Decisión proyecto		Revisión estándar	Revisión/aprobación por los miembros del proyecto	Según establecimiento del equipo.
2.2	Organización y roles	Estándar PMBOK	Revisión estándar	Revisión/aprobación por los miembros del proyecto	Según estándar.
2.3	Identificación requisitos	Norma IEEE 830	Revisión estándar	Revisión/aprobación por los miembros del proyecto	Según estándar.
2.4	Elaboración documentación	Estándar PMBOK	Revisión estándar	Revisión/aprobación por los miembros del proyecto	Según estándar.
2.5	Adquisición licencias		Revisión estándar	Revisión/aprobación por los miembros del proyecto	<ul style="list-style-type: none"> <li>● Licencias adquiridas de: <ul style="list-style-type: none"> <li>○ 3dsmax</li> <li>○ Visual Studio</li> <li>○ Unity</li> <li>○ FL Studio</li> <li>○ Pro Tools</li> <li>○ Photoshop</li> </ul> </li> </ul>
3.1.1	Concept Art Interfaz		Revisión estándar	Revisión/aprobación por los miembros del equipo de arte y PM	<ul style="list-style-type: none"> <li>● Concept art formato digital.</li> </ul>
3.1.2	Elaboración assets interfaz	Estándar ISO 13407	Revisión estándar	Revisión/aprobación por los miembros del equipo de arte y PM	<ul style="list-style-type: none"> <li>● Uso de canvas en Unity.</li> <li>● Resolución 1080p.</li> <li>● Responsive, adaptables a distintas configuraciones.</li> </ul>

3.2.1	Concept Art Assets		Revisión estándar	Revisión/aprobación por los miembros del proyecto	<ul style="list-style-type: none"> <li>● Concept art formato digital.</li> <li>● Paleta de colores definida.</li> <li>● Estilo visual definido.</li> </ul>
3.2.2	Modelado y texturizado		Revisión estándar	Revisión/aprobación por los miembros del equipo de arte y PM	<ul style="list-style-type: none"> <li>● Modelos 3D en formato .fbx</li> <li>● Texturas sencillas, potencia de 2, en la paleta de colores establecida.</li> </ul>
3.2.3	Rigging y animación		Revisión estándar	Revisión/aprobación por los miembros del equipo de arte y PM	<ul style="list-style-type: none"> <li>● Rigging sistema CAT.</li> <li>● Bucles animación de acciones rutinarias establecidas.</li> <li>● Animaciones estáticas por keyframes.</li> </ul>
3.2.4	Assets finales		Revisión estándar	Revisión/aprobación por los miembros del proyecto	<ul style="list-style-type: none"> <li>● Assets organizados siguiendo código nomenclatura.</li> <li>● Mismas características de exportación.</li> </ul>
4.1	Programación físicas	ISO/IEC 25000	Revisión estándar	Revisión/aprobación por los miembros del equipo de programación y PM	<ul style="list-style-type: none"> <li>● Base físicas establecida usando bounding boxes y colliders simples.</li> <li>● Uso motor Unity.</li> <li>● Uso componentes predefinidos Unity.</li> </ul>
4.2	Programación mecánicas	ISO/IEC 25000	Revisión estándar	Revisión/aprobación por los miembros del equipo de programación y PM	<ul style="list-style-type: none"> <li>● Mecánicas con entrada de usuario compatible desde distintos dispositivos.</li> <li>● Empleo paquete Cinemachine.</li> <li>● Código escalable y modularizado.</li> </ul>
4.3	Programación IA	ISO/IEC 25000	Revisión estándar	Revisión/aprobación por los miembros del equipo de programación y PM	<ul style="list-style-type: none"> <li>● Sistema IA establecido a través de navmeshes para el comportamiento de personajes.</li> <li>● Comportamiento implementado definido por máquinas de estados.</li> <li>● Algoritmos de baja complejidad, código legible.</li> </ul>
4.4	Integración arte		Revisión estándar	Revisión/aprobación por los miembros del proyecto	<ul style="list-style-type: none"> <li>● Aplicación sin placeholders y con los assets finales integrados en nuestro proyecto de Unity.</li> </ul>
5.1	Producción sonido		Revisión estándar	Revisión/aprobación por los miembros del equipo de sonido y PM	<ul style="list-style-type: none"> <li>● Archivos finales .wav para producciones musicales .mp3 para efectos y sonidos cortos.</li> </ul>

5.2	Composición música		Revisión estándar	Revisión/aprobación por los miembros del equipo de sonido y PM	<ul style="list-style-type: none"> <li>Archivos finales .wav para producciones musicales .mp3 para efectos y sonidos cortos.</li> <li>Frecuencia de muestreo y profundidad de bits: 48 KHz y 24 bits.</li> </ul>
5.3	Integración sonido		Revisión estándar	Revisión/aprobación por los miembros del equipo de sonido, programación y PM	<ul style="list-style-type: none"> <li>Archivos finales .wav para producciones musicales .mp3 para efectos y sonidos cortos.</li> <li>Frecuencia de muestreo y profundidad de bits: 48 KHz y 24 bits.</li> </ul>
6.1	Testing interfaz		Revisión estándar	Revisión/aprobación por los miembros del proyecto	Según contrato y requisitos.
6.2	Testing juego		Revisión estándar	Revisión/aprobación por los miembros del proyecto	Según contrato y requisitos.
6.3	Testing adaptabilidad		Revisión estándar	Revisión/aprobación por los miembros del proyecto	Según contrato y requisitos.
7.1	Adquisición de licencias y distribución		Revisión estándar	Revisión/aprobación por los miembros del proyecto	Licencias adquiridas en las distintas plataformas acordadas.
7.2	Material Publicado		Revisión estándar	Revisión/aprobación por los miembros del proyecto	Material publicado en las distintas plataformas.

## ROLES PARA LA GESTIÓN DE LA CALIDAD

En este apartado planteamos los roles que llevarán a cabo el control de calidad del proyecto, así como sus funciones y responsabilidades. También adjuntamos un pequeño organigrama para poder visualizar de manera rápida los encargados de la supervisión de la calidad en el proyecto.

Jerarquía rol	Nombre	Responsabilidades
1º	Project Manager	<ul style="list-style-type: none"> <li>● Responsable: Denis.</li> <li>● Funciones rol: asegurar cumplimiento de estándares y entregables. Supervisar, aprobar y tomar acciones correctivas en caso de ser necesario.</li> <li>● Supervisa a: resto de miembros de proyecto.</li> <li>● Niveles de autoridad: exigir cumplimiento de entregables.</li> <li>● Requisitos de habilidades: organización, liderazgo, motivación, mediación de conflictos, comunicación, visión general.</li> </ul>
2º	Resto de miembros proyecto	<ul style="list-style-type: none"> <li>● Funciones rol: elaboración de los entregables de acuerdo con lo establecido previamente.</li> <li>● Supervisa a: cada departamento tendrá un especial control llevado a cabo por sus propios integrantes.</li> <li>● Niveles de autoridad: aplicar los recursos que se han asignado para desarrollar los entregables.</li> <li>● Requisitos de habilidades: especificadas según los entregables.</li> </ul>

Aunque la autoridad responsable de los entregables en última instancia sea el Project manager dependiendo del campo al que referencia cada paquete el equipo pertinente tendrá un mayor papel en el control de calidad que otros equipos ajenos a dicha producción.



