***estructura análisis***

A tener en cuenta: la documentación es la presentacion exhaustiva de info sobre la metodología y solo la metodología, cada detalle.

En cambio la carpeta es la info mas rasca y sin presentacion: miembros y roles, qué metodologia se usa y pq creemos es la mejor para nosotros, muestras del proyecto y los objetivos del mismo, ademas de un analisis. (mucha atencion al analisis y como se hace). /a continuacion poner elementos dictados a tener en cuenta)

***RAD***

***(Rapid Application Development)***

***¿Cuándo surge esta metodología***

***y el razonamiento que propone? (origen)***

~~~jugar con esto en clase, en lo posible rellenar con algunas imágenes y ejemplos~~~

-Año de origen

-Desarrolladores(sitepinta)

-Qué vino a resolver

-Para qué contextos es la mejor opción

***¿En qué se basa y cómo se fundamenta?***

***se basa en: (desarrollá todas al menos por encima)***

**-Desarrollo Iterativo**

**-Prototipado funcional**

**-Participación constante del usuario**: (Garantiza que se esté haciendo lo que se desea realmente como producto final.

**-Interdisciplinacion**: (equipos y/o individuos que se especializan en varias areas)

**-Autonomia y Colaboracion**; si bien todos colaboran entre si, es clave que se tenga la capacidad de ser autonomo para adaptarse a cualquier cambio sin perder la calidad del desarrollo.

**-Reciclado**: Reutilización de elementos existentes que no haga falta rehacer de 0, permitiendonos ahorrar tiempo.

***Cuales son sus valores?***

**-Flexibilidad** y Adaptabilidad:(respuesta rápida a los cambios, contextos y adaptabilidad frente a cambios de último momento, tanto en requisitos como en el producto final)

**Escalabilidad:** La posibilidad y el cómo escalar la metodología para proyectos de diferentes tamaños y complejidades según también, las necesidades. (se relaciona muchísimo con la flexibilidad)

**-Tiempos de desarrollo cortos**:(entrega rápida de prototipos)

**-Satisfacción del Cliente**:La satisfacción del cliente es un objetivo principal, logrado a través de la entrega continua del software funcional de valor y la adaptación a sus comentarios.

**-Entrega de Alta Calidad**: A pesar de la rapidez del desarrollo, se mantiene un alto estándar de calidad mediante pruebas continuas y refinamiento iterativo de los prototipos.

**-Optimización del Proceso**: Se optimiza el proceso de desarrollo para maximizar la eficiencia y minimizar el desperdicio de recursos.

-**Transparencia**: La comunicación clara y abierta entre todos los miembros del equipo y con los usuarios es esencial para el éxito del proyecto.

***¿Cuáles son sus componentes?***

-**Fases de desarrollo:**

**-Roles y responsabilidades:**

***-Documentación:***

***¿Cómo se aplica?***

***y sus Buenas Prácticas***

**-Aplicación paso a paso(con ejemplos de casos reales, si o si):**

**-Herramientas como**:

1)Sistemas de Gestión de Contenidos.

2)Colaboración y Control de Versiones (Github ponele)

3)Plataformas de Documentación: Plataformas específicas para documentación como ReadTheDocs o Docusaurus.

**-Buenas Prácticas/recomendadas:**

***Evaluación y Mejora Continua***

-Indicadores de Desempeño: Define métricas e indicadores para evaluar la eficiencia y correcto desempeño/funcionamiento de la metodología.

-Retroalimentación y Ajustes: Proporciona un marco para la retroalimentación continua y ajuste de la metodología.

***Formato y Accesibilidad***

**-Estructura Clara y Navegación**: Asegura que la documentación esté bien organizada y sea fácil de navegar.

-**Formato Digital:** Utiliza un formato que sea accesible y legible en múltiples dispositivos (e.g., HTML, PDF).

Es una metodología que, misma toma en sí, elementos de otras metodologías como(…) y se basa en (los métodos de desarrollo, tipo iteración y todas esas cagadas) a base del desarrollo, en un primer lugar, de un prototipo que toma en cuenta cosas como (objetos, mecánicas, coexistencia de elementos, viabilidad, tiempos, capacidades de trabajo etc ) y ayudan a asimilar la posibilidad de aplicar o descartar dichos elementos dentro del que sería en un futuro el desarrollo del producto final.

### Metodología RAD (Rapid Application Development) para tu Proyecto de Videojuego

#### Características Clave de RAD:

1. \*\*Ciclos de Desarrollo Cortos:\*\*

- Los ciclos de desarrollo en RAD son cortos y enfocados en la entrega rápida de prototipos y componentes funcionales.

- Estos ciclos permiten realizar ajustes rápidos basados en la retroalimentación continua de los usuarios y el equipo.

2. \*\*Prototipos Rápidos:\*\*

- Se crean prototipos rápidos para experimentar con diferentes ideas y obtener retroalimentación temprana.

- Los prototipos ayudan a clarificar los requisitos y asegurar que el producto final cumpla con las expectativas.

3. \*\*Equipos Multidisciplinarios:\*\*

- Los equipos en RAD suelen ser pequeños, altamente cualificados y multidisciplinarios, lo que permite una colaboración eficiente y efectiva.

- La participación activa de los usuarios finales es clave para asegurar que el desarrollo esté alineado con sus necesidades.

4. \*\*Iteración Continua:\*\*

- Cada ciclo de desarrollo incluye planificación, diseño, implementación y pruebas, permitiendo mejoras continuas y entrega incremental del producto.

### Implementación de RAD para tu Proyecto de Videojuego

#### Fases y Actividades

1. \*\*Planificación Inicial (Primera Semana):\*\*

- \*\*Definición del Proyecto:\*\* Clarifica los géneros del videojuego y establece los objetivos principales.

- \*\*Reunión de Ideas:\*\* Realiza sesiones de brainstorming para generar ideas sobre la historia, mecánicas, gráficos y sonido.

- \*\*Cronograma de Ciclos Cortos:\*\* Define un cronograma para los ciclos de desarrollo y las entregas bimensuales.

2. \*\*Ciclos de Desarrollo Cortos (Ciclos de 2 Semanas):\*\*

- \*\*Primera Iteración:\*\*

- \*\*Prototipos Iniciales:\*\* Desarrolla prototipos rápidos de las mecánicas clave del juego.

- \*\*Retroalimentación Temprana:\*\* Obtén retroalimentación del equipo y, si es posible, de usuarios potenciales.

- \*\*Informe y Presentación:\*\* Prepara y entrega un informe detallando los avances y las retroalimentaciones recibidas.

- \*\*Iteraciones Subsiguientes:\*\*

- \*\*Desarrollo Incremental:\*\* En cada iteración, añade nuevas funcionalidades y mejora las existentes. Cada ciclo debe incluir planificación, diseño, implementación y pruebas.

- \*\*Evaluación y Ajuste:\*\* Basado en la retroalimentación, ajusta el desarrollo del videojuego.

- \*\*Informes Bimensuales:\*\* Al final de cada ciclo de dos semanas, entrega un informe y presenta los avances, destacando las mejoras y cambios realizados.

3. \*\*Evaluación Continua y Prototipado (Duración Continua):\*\*

- \*\*Mejora de Usabilidad:\*\* Usa los prototipos para probar la jugabilidad y realizar ajustes basados en la experiencia del usuario.

- \*\*Retroalimentación del Usuario:\*\* Mantén una línea continua de retroalimentación de los usuarios finales, ajustando y refinando el juego en cada iteración.

4. \*\*Fase de Pulido (Últimas 4 Semanas):\*\*

- \*\*Optimización y Corrección de Errores:\*\* Enfócate en la optimización del rendimiento del juego y la corrección de errores críticos.

- \*\*Pruebas Finales:\*\* Realiza pruebas beta con un grupo más amplio de usuarios, recopilando retroalimentación final.

- \*\*Preparación para la Entrega:\*\* Completa la documentación del proyecto y prepara la presentación final para el colegio.

### Beneficios de RAD para tu Proyecto

- \*\*Rapidez en el Desarrollo:\*\* Los ciclos cortos y la entrega rápida de componentes funcionales aseguran un progreso constante.

- \*\*Flexibilidad y Adaptación:\*\* RAD permite adaptarse rápidamente a los cambios en los requisitos y la retroalimentación de los usuarios.

- \*\*Participación Activa de los Usuarios:\*\* La retroalimentación continua de los usuarios finales garantiza que el producto final cumpla con sus expectativas.

- \*\*Cumplimiento de Plazos:\*\* La estructura de ciclos cortos se alinea perfectamente con la necesidad de informes y presentaciones bimensuales.

### Cronograma Tentativo con RAD

| Semana | Actividad Principal | Entrega |

|--------|----------------------------------------------------------|---------|

| 1 | Planificación Inicial | - |

| 2 | Prototipo de Mecánicas Clave, Retroalimentación | Informe 1 |

| 4 | Desarrollo Incremental, Ajuste Basado en Retroalimentación| Informe 2 |

| 6 | Nuevas Funcionalidades, Evaluación Continua | Informe 3 |

| 8 | Mejoras y Refinamientos, Retroalimentación | Informe 4 |

| 10 | Desarrollo Incremental, Evaluación Continua | Informe 5 |

| 12 | Nuevas Funcionalidades, Prototipo de Gráficos/Sonido | Informe 6 |

| 14 | Ajustes y Mejoras, Retroalimentación | Informe 7 |

| 16 | Optimización y Corrección de Errores | Informe 8 |

| 18 | Pruebas Finales, Retroalimentación | Informe 9 |

| 20 | Preparación para la Entrega Final | Informe 10 |

| 21-24 | Pulido Final y Documentación | - |

### Conclusión

La metodología RAD ya incorpora las mejores prácticas de prototipado rápido y desarrollo iterativo, lo que la hace muy adecuada para tu proyecto de videojuego. Este enfoque te permitirá avanzar de manera rápida y flexible, adaptándote a la retroalimentación continua y asegurando que cumples con las entregas e informes bimensuales.

CARPETA TÉCNICA

### 1. \*\*Definir la Visión y el Alcance del Proyecto\*\*

- \*\*Visión del Juego:\*\* Describan en una frase o dos el concepto del juego, su estilo y el público objetivo.

- \*\*Características Principales:\*\* Enumeren las características esenciales del juego. Mantengan el alcance manejable para evitar el "scope creep".

- \*\*Plataformas:\*\* Decidan en qué plataformas se lanzará el juego (PC, consolas, móviles).

### 2. \*\*Roles y Responsabilidades\*\*

Asignen roles específicos a cada miembro del equipo según sus habilidades:

- \*\*Líder de Proyecto (tú):\*\* Gestión del proyecto, comunicación, seguimiento de plazos y coordinación general.

- \*\*Programador:\*\* Desarrollo de la lógica del juego, implementación de mecánicas y solución de problemas técnicos.

- \*\*Artista:\*\* Diseño de personajes, entornos, UI, y otros elementos visuales.

- \*\*Diseñador de Niveles:\*\* Creación de niveles, balance del juego y ajuste de la jugabilidad.

- \*\*Músico/Compositor:\*\* (Si aplica) Creación de la música y efectos de sonido (puede ser un rol compartido o externo).

### 3. \*\*Herramientas y Recursos\*\*

- \*\*Motor de Juego:\*\* Seleccionen un motor de juego (Unity, Unreal Engine, Godot, etc.).

- \*\*Gestión de Proyectos:\*\* Utilicen herramientas como Trello, Asana o Jira para organizar tareas y plazos.

- \*\*Control de Versiones:\*\* Implementen un sistema de control de versiones como GitHub para gestionar el código fuente.

- \*\*Comunicación:\*\* Usen herramientas de comunicación como Discord o Slack para mantener una comunicación fluida.

### 4. \*\*Planificación del Proyecto\*\*

- \*\*Roadmap del Proyecto:\*\* Crear un cronograma general del proyecto con hitos importantes (prototipo, alfa, beta, lanzamiento).

- \*\*División en Sprints:\*\* Dividir el trabajo en sprints (períodos cortos de trabajo, generalmente 1-2 semanas) con metas específicas.

- \*\*Reuniones Regulares:\*\* Programar reuniones regulares (semanales o bisemanales) para revisar el progreso, resolver problemas y ajustar el plan.

### 5. \*\*Desarrollo Iterativo\*\*

- \*\*Prototipo Inicial:\*\* Crear un prototipo jugable lo antes posible para probar las mecánicas básicas.

- \*\*Feedback y Pruebas:\*\* Hacer pruebas frecuentes y recolectar feedback para mejorar el juego continuamente.

- \*\*Iteración:\*\* Refinar y mejorar el juego en ciclos iterativos, añadiendo características y puliendo detalles.

### 6. \*\*Marketing y Comunidad\*\*

- \*\*Presencia en Redes Sociales:\*\* Crear y mantener perfiles en redes sociales (Twitter, Instagram, YouTube, etc.) para atraer y mantener el interés del público.

- \*\*Devlog:\*\* Publicar actualizaciones regulares sobre el desarrollo del juego en un blog o foro.

- \*\*Demo y Beta Testing:\*\* Lanzar una demo y reclutar beta testers para recibir feedback y detectar bugs.

### 7. \*\*Lanzamiento y Post-Lanzamiento\*\*

- \*\*Plan de Lanzamiento:\*\* Definir una estrategia de lanzamiento (fechas, plataformas, precios).

- \*\*Soporte Post-Lanzamiento:\*\* Planificar actualizaciones y soporte después del lanzamiento para corregir bugs y añadir contenido adicional.

### 8. \*\*Documentación\*\*

- \*\*Documentación del Proyecto:\*\* Mantener una documentación detallada del diseño del juego, código, arte y cualquier otro aspecto relevante.

- \*\*Manual del Jugador:\*\* Crear un manual del jugador o tutorial dentro del juego para guiar a los nuevos jugadores.

### Ejemplo de Cronograma Simplificado

- \*\*Mes 1:\*\* Definición del proyecto, roles y herramientas, prototipo inicial.

- \*\*Meses 2-3:\*\* Desarrollo de las mecánicas principales, creación de arte inicial, diseño de niveles.

- \*\*Meses 4-5:\*\* Pruebas y refinamiento del prototipo, integración de música y efectos.

- \*\*Meses 6-7:\*\* Desarrollo avanzado, marketing inicial, preparación para la beta.

- \*\*Mes 8:\*\* Beta testing, ajustes finales.

- \*\*Mes 9:\*\* Preparación del lanzamiento, marketing intensivo.

- \*\*Mes 10:\*\* Lanzamiento, inicio del soporte post-lanzamiento.

### Conclusión

Mantén la comunicación abierta y asegúrate de que todos los miembros del equipo estén alineados con los objetivos del proyecto. La clave es la planificación, la organización y la flexibilidad para adaptarse a los cambios y desafíos que surjan en el camino. ¡Buena suerte con tu proyecto de videojuego!

Entendido, siendo un proyecto escolar, el enfoque será más simple y ajustado a los plazos académicos. Aquí tienes una guía adaptada a un entorno educativo:

### 1. \*\*Definir el Proyecto\*\*

- \*\*Objetivo del Juego:\*\* Describir brevemente el concepto del juego y su objetivo.

- \*\*Requisitos del Proyecto:\*\* Asegúrense de comprender los requisitos del proyecto escolar (plazos, criterios de evaluación, entregables).

### 2. \*\*Roles y Responsabilidades\*\*

Asignen roles específicos según las habilidades de cada miembro y la necesidad del proyecto:

- \*\*Líder del Proyecto (tú):\*\* Coordinación general, gestión del tiempo y comunicación con el profesor.

- \*\*Programador:\*\* Desarrollo del código y lógica del juego.

- \*\*Artista:\*\* Creación de gráficos, personajes y escenarios.

- \*\*Diseñador de Niveles:\*\* Planificación y diseño de los niveles del juego.

(En un equipo escolar, los roles pueden superponerse y todos pueden colaborar en diferentes aspectos según sea necesario.)

### 3. \*\*Planificación del Proyecto\*\*

- \*\*Cronograma:\*\* Crear un cronograma con fechas importantes y plazos específicos.

- \*\*Tareas Principales:\*\*

- Investigación y planificación inicial.

- Desarrollo del prototipo.

- Creación de assets (gráficos, sonido).

- Implementación de niveles y mecánicas.

- Pruebas y refinamientos.

- Preparación de la presentación final.

### 4. \*\*Herramientas y Recursos\*\*

- \*\*Motor de Juego:\*\* Utilicen un motor accesible y apropiado para el nivel escolar, como Unity o Godot.

- \*\*Gestión de Tareas:\*\* Usen una herramienta sencilla como Trello para organizar y asignar tareas.

- \*\*Control de Versiones:\*\* GitHub puede ser útil, pero para proyectos escolares, mantener una copia de seguridad en la nube (Google Drive, Dropbox) puede ser suficiente.

- \*\*Comunicación:\*\* Utilicen una plataforma de comunicación como WhatsApp o Discord.

### 5. \*\*Desarrollo Iterativo\*\*

- \*\*Prototipo Inicial:\*\* Desarrollen un prototipo básico rápidamente para tener una base sobre la que trabajar.

- \*\*Pruebas Regulares:\*\* Realicen pruebas frecuentes para detectar y solucionar problemas.

- \*\*Iteración:\*\* Mejoren el juego de manera continua, añadiendo características y refinando detalles en ciclos cortos.

### 6. \*\*Presentación del Proyecto\*\*

- \*\*Documentación:\*\* Mantengan una documentación clara de todas las partes del proyecto, incluyendo el diseño, el código y los gráficos.

- \*\*Demostración:\*\* Preparar una demostración del juego que puedan presentar a la clase o al profesor.

- \*\*Informe:\*\* Redacten un informe final que explique el desarrollo del proyecto, las dificultades encontradas y cómo las superaron.

### Ejemplo de Cronograma Simplificado

- \*\*Semana 1:\*\* Definición del proyecto y asignación de roles.

- \*\*Semana 2:\*\* Investigación y desarrollo del prototipo inicial.

- \*\*Semana 3:\*\* Desarrollo de las mecánicas principales y creación de assets gráficos.

- \*\*Semana 4:\*\* Diseño e implementación de los primeros niveles.

- \*\*Semana 5:\*\* Pruebas y refinamiento del juego.

- \*\*Semana 6:\*\* Preparación de la presentación final y documentación.

### Conclusión

En un proyecto escolar, la clave es la organización y la colaboración. Mantén una comunicación abierta y asegúrate de que todos los miembros del equipo estén involucrados y al tanto de sus responsabilidades. Cumplir con los plazos y estar preparados para la presentación final será crucial para el éxito del proyecto. ¡Buena suerte!