}

**Guía1. Definición Proyecto APT**

**Asignatura Capstone**

1. **PARTE I**

|  |
| --- |
| **1. Antecedentes Personales** |
| A continuación, se presenta una tabla en la que debes completar la información solicitada. |

|  |  |
| --- | --- |
| Nombre estudiante | **Benjamín Polanco** |
| Rut | **20.220.892-4** |
| Carrera | **Ingeniería en Informática** |
| Sede | **Maipú** |

|  |
| --- |
| **2. Descripción Proyecto APT** |
| En la descripción debes señalar brevemente el nombre de tu proyecto APT y las competencias del perfil de egreso que vas a poner en práctica. Si en tu carrera están definidas las áreas de desempeño, también menciona a qué áreas de desempeño está vinculado el proyecto. |

|  |  |
| --- | --- |
| Nombre del proyecto | **Offlimits** |
| Área (s) de desempeño(s) | * **Programación de Algoritmos.** * **Desarrollo de Software.** * **Calidad de Software.** * **Integración de Plataformas.** * **Emprendimiento.** |
| Competencias | * **Desarrollo de Soluciones de Software.** * **Integración de Plataformas.** * **Gestión de Requerimientos.** * **Gestión de Proyectos Informáticos.** * **Gestión de Calidad.** * **Gestión de Riesgos.** |

|  |
| --- |
| **3. Fundamentación Proyecto APT** |
| A continuación, se presentan distintos campos que debes completar con la información solicitada. Esta sección busca que describas en detalle tu proyecto y justifiques su relevancia y pertinencia. |

|  |  |
| --- | --- |
| Relevancia del proyecto APT | “Offlimits” es una modificación para el videojuego Minecraft, cual fue creado el año 2009 por el desarrollador sueco Markus Persson (Notch) y posteriormente adquirido por Microsoft el año 2014, fue creado como un juego tipo **SandBox**, que permite al usuario tener libertad absoluta sobre el juego, otorgándole las herramientas para crear y destruir a su antojo.  Minecraft fue lanzado oficialmente el año 2011 y se ha estado actualizando desde entonces por medio de versiones semánticas, estas funcionan en base a una serie de números, uno que cataloga su versión de lanzamiento, seguido por uno que cataloga una actualización mayor, y finalmente uno que cataloga una actualización menor como corrección de errores (ej. 1.12.2).  Cada actualización incluye nuevo contenido para mantener cautivado al usuario, y adicionalmente existe una comunidad que crea contenido de manera no oficial que los usuarios pueden integrar a su juego (modificaciones o “**mods**”).  Para poder instalar estas modificaciones, la comunidad creó unos “lanzadores” que permiten al juego ejecutar estos mods, los lanzadores más populares son **Forge** y **Fabric**, que, a pesar de cumplir los mismos objetivos, son incompatibles entre sí y los mods deben ser desarrollados utilizando las herramientas que estos lanzadores proveen.  Esto lleva a los mods verse afectados por la diversidad entre lanzadores, mientras Forge tiene mayor variedad, Fabric tiene mejor rendimiento, por lo que algunos desarrolladores optan por hacer el mod dos veces, uno en cada lanzador.  Los mods también se ven afectados por las actualizaciones, por lo cual algunos son creados para versiones específicas o se van actualizando a la par que el juego base, sin embargo, algunos desarrolladores optan por actualizar hasta cierta versión antes de descontinuar el mod.  Minecraft contaba con un límite de altura de 256 bloques, muchos usuarios solicitaban un límite de altura mayor y los desarrolladores de mods estaban al tanto de esto, pero no hubo mucho enfoque al respecto ya que era una tarea complicada.  En la actualización 1.17, Minecraft integró un cambio de altura de 256 a 320 bloques, sin embargo, también logró agregar 64 bloques por debajo de la altura mínima, creando así la ilusión de un mundo más profundo.  Dicha actualización también integró nuevos cambios que afectó negativamente el rendimiento en ordenadores más antiguos, por lo que algunos usuarios optaron por quedarse en la versión 1.16, que, a pesar de no tener estos cambios, era más estable.  Entre otras razones por la cual estos usuarios decidieron quedarse es porque algunos mods no se actualizaron más allá de esta versión y querían mantenerlos en sus partidas.  Existen algunos mods que agregan contenido de actualizaciones futuras a esta versión, permitiendo a estos usuarios el disfrutar de nuevo contenido sin tener que comprometer sus partidas. |
| Descripción del Proyecto APT | Offlimits busca introducir un cambio en la altura del mundo en Minecraft para la versión 1.16, adicionalmente introduciría cambios en la generación de terreno para introducir cuevas más amplias y profundas.  El mod contaría con un sistema de configuraciones para darle al usuario una experiencia más personalizada.  Finalmente, este mod seria desarrollado tanto como para los lanzadores Forge y Fabric, para cumplir con la demanda por ambas comunidades. |
| Pertinencia del proyecto con el perfil de egreso | * **Desarrollo de Soluciones de Software:**   El proyecto requiere trabajar en características internas del juego, en otras palabras, modificar como Minecraft genera el terreno.  A su vez, dado a que este proyecto se enfoca en una versión anterior, este debe estar bien optimizado para no afectar el rendimiento de los usuarios con la carga adicional de terreno y límite de altura.   * **Integración de Plataformas:**   Al estar desarrollando este tipo de proyectos, se nos da a elegir entre 2 principales plataformas que permiten al usuario ejecutar el juego con modificaciones, cada una con sus herramientas únicas.  Adicionalmente en la mayoría de los casos, se recomienda a los desarrolladores integrar distintas APIs a sus proyectos por medio de **Gradle** para facilitar el desarrollo.   * **Gestión de Requerimientos:**   Este proyecto se llevó a cabo por las solicitudes de la comunidad, muchos usuarios solicitaban una modificación que llevase a cabo estas características, lo que llevó a tomar los requerimientos en base a las necesidades de los usuarios, la compatibilidad, el rendimiento, y la personalización.   * **Gestión de Proyectos Informáticos:**   El proyecto debe estar bien planificado para llevarse a cabo con la menor cantidad de inconvenientes posibles, identificando cuales modificaciones populares podrían ocasionar conflicto y adaptar el proyecto para que este sea compatible con los más destacables.   * **Gestión de Calidad:**   El proyecto debe realizar variadas pruebas para asegurar que este cumpla con los requerimientos especificados, y que este funcione correctamente en los entornos tanto de cliente como de servidor, y que no interfiera con otras modificaciones instaladas por el usuario.  Adicionalmente hay que realizar pruebas de rendimiento para que el proyecto no afecte negativamente al usuario.   * **Gestión de Riesgos:**   El proyecto cuenta con características con gran potencial de romper la compatibilidad con otras modificaciones, por lo cual se tendrá que realizar pruebas para identificar con cuáles modificaciones habrá que tener consideración especial para agregar compatibilidad o marcar como incompatibles hasta una futura actualización.  A su vez, es posible que el proyecto al aumentar la cantidad de bloques por generar en el terreno llegue a sobrecargar el sistema y bajar el rendimiento en ordenadores menos potentes.  Finalmente hay que tener en cuenta que el proyecto puede no ser aceptado por la comunidad por no cumplir todas sus necesidades o porque decidieron mudarse a versiones futuras sin importar el riesgo del rendimiento. |
| Relación con los intereses profesionales | Entre mis intereses se encuentra el desarrollo de videojuegos, me interesa el tema de modificación de videojuegos ya que nos permite trabajar directamente desde el código fuente del videojuego.  Me interesa trabajar en este proyecto ya que muchos desarrolladores de videojuegos en la industria comenzaron modificando y han sido contratados por las empresas dueñas del videojuego modificado. |
| Factibilidad de desarrollo del Proyecto APT | Es posible desarrollar el proyecto durante el semestre dado a que es un proyecto que estoy desarrollando fuera de las horas asignadas a la asignatura y ya cuenta con un porcentaje desarrollado.  Los materiales que se necesitan se pueden obtener de manera gratuita en internet, ya que trabajamos con IntelliJ como nuestra IDE, y dentro del proyecto utilizamos **Java**, **Gradle**, **Git**, **Json**, **Maven** y **Mixins**.  He desarrollado una API que me facilita el desarrollo entre distintas plataformas por lo cual no hay que gastar tiempo en pasar el proyecto de una plataforma a otra.  Lo único que podría dificultar su desarrollo es la incompatibilidad con otras modificaciones, sin embargo, se podría identificar cuáles modificaciones son esenciales para ser compatibles (tales como de optimización del juego) y cuales podemos delegar para una actualización futura. |

1. **PARTE II**

|  |
| --- |
| **4. Objetivos** |
| En este apartado debes definir objetivos generales y específicos del Proyecto APT. Es importante aclarar que los objetivos se deben plantear en forma clara, concisa y sin dar mayores explicaciones, es decir, deben entenderse por sí solos. Se sugiere redactarlos utilizando un verbo en infinitivo, pues ello obliga a precisar acciones concretas. |

|  |  |
| --- | --- |
| Objetivo general | El proyecto debe incrementar el límite de altura tanto positivo como negativo en Minecraft 1.16.  Adicionalmente debe incluir generación de terreno en aspecto a la altura mínima y una generación de cuevas más profundas,  Finalmente debe de tener configuraciones para personalizar la experiencia del usuario en base a la altura o generación del terreno. |
| Objetivos específicos | * **Toma de requerimientos.** * **Diseño de arquitectura de software.** * **Implementación de nuevo límite de altura.** * **Implementación de nueva generación de terreno.** * **Implementación de entidades.** * **Implementación de compatibilidad.** * **Optimización.** * **Lanzamiento y Monitoreo.** |

|  |
| --- |
| **5. Metodología** |
| En el siguiente apartado deberás describir la metodología, propia de tu disciplina, que utilizarás para resolver el proyecto APT antes descrito, incluyendo las etapas y métodos de trabajo. |

|  |
| --- |
| Descripción de la Metodología |
| **Para este proyecto, se optó por utilizar una metodología en cascada debido a su enfoque estructurado, secuencial y controlado, y por la necesidad de seguir un enfoque ordenado para cumplir con los objetivos específicos de manera eficiente.**   1. **Fase de Requisitos:**  * **Objetivo:** Definir los cambios que se harán en la modificación para cumplir con los objetivos específicos. * **Acciones:**   + **Toma de requerimientos en cambios en las alturas.**   + **Toma de requerimientos en generación del terreno.**   + **Identificación de modificaciones esenciales para compatibilidad.** * **Entregable:** Documento que incluye los requerimientos funcionales para los objetivos específicos.  1. **Fase de Diseño del Sistema:**  * **Objetivo:** Diseñar los componentes que soportarán las nuevas características del proyecto. * **Acciones:**   + **Diseño de componentes que manejarán la nueva altura.**   + **Diseño de componentes que ayudarán a las herramientas de depuración.**   + **Diseño de componentes que manejarán la nueva generación de terreno.**   + **Diseño de componentes que manejarán la generación de entidades.**   + **Diseño de componentes que manejarán la compatibilidad entre otras modificaciones.**   + **Revisión de componentes desarrollados en busca de posibles optimizaciones.** * **Entregable:** Documentación técnica que detalle la implementación de los objetivos específicos del proyecto.  1. **Fase de Implementación:**  * **Objetivo:** Desarrollar los objetivos específicos del proyecto. * **Acciones:**   + **Creación del ambiente de trabajo.**   + **Incrementación de altura por coordenadas positivas y negativas.**   + **Actualización de herramientas de depuración.**   + **Actualización del renderizado del mundo.**   + **Integración de la nueva generación de terreno.**   + **Integración de entidades en coordenadas negativas.**   + **Integración de compatibilidad con modificaciones esenciales seleccionadas.** * **Entregable:** El proyecto como un producto funcional, incluyendo todas las características mencionadas en los objetivos específicos.  1. **Fase de Pruebas:**  * **Objetivo:** Realizar pruebas exhaustivas para asegurarse que el proyecto cumpla todos los requerimientos definidos. * **Acciones:**   + **Pruebas con la construcción sobre coordenadas positivas y negativas.**   + **Pruebas con la generación de terreno.**   + **Pruebas de inteligencia artificial y navegación de entidades.**   + **Pruebas de compatibilidad.**   + **Pruebas de rendimiento.** * **Entregable:** Documento de pruebas que informe los resultados y posibles problemas encontrados, con sus respectivas soluciones.  1. **Fase de Mantenimiento:**  * **Objetivo:** Realizar un monitoreo al proyecto tras su lanzamiento, estando en contacto con la comunidad. * **Acciones:**   + **Monitoreo de la retroalimentación.**   + **Monitoreo de los problemas notificados en GitHub.**   + **Actualización del proyecto en casos catastróficos.** * **Entregable:** Actualizaciones y parches, junto a un registro de cambios documentado. |

|  |
| --- |
| **6. Evidencias** |
| A continuación, describe qué evidencias serán evaluadas en el informe de avance y en el informe final de tu proyecto APT. Estas evidencias deben ser acordadas con tu docente. Se entenderá por evidencia los productos que se desarrollen durante el proyecto y cuyo propósito sea visibilizar o documentar cómo se ha implementado el trabajo. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de evidencia**  **(avance o final)** | **Nombre de la evidencia** | **Descripción** | **Justificación** |
| **Avance** | **Informe de Requerimientos.** | **Documento que incluye lista de los requerimientos funcionales y no funcionales para los objetivos específicos definidos para el proyecto.** | **Este documento presenta cómo se han identificado y definido los objetivos del proyecto.** |
| **Avance** | **Informe del Diseño Técnico del Sistema.** | **Documento que incluye diagramas de arquitectura para los componentes y especificaciones de cómo se implementan las funciones del proyecto.** | **Este documento evidencia la planificación y arquitectura técnica del producto, asegurando que el desarrollo sigue un diseño estructurado.** |
| **Avance** | **Informe del Proceso de Implementación.** | **Reporte que describe el progreso actual en la implementación de los objetivos específicos, informando lo que se ha implementado y lo que falta por implementar.** | **Este documento proporciona una visión general del estado actual del desarrollo, dando evidencia de las funcionalidades ya completadas.** |
| **Avance** | **Producto Funcional.** | **Versión funcional del producto que incluye todas las funciones especificadas para su desarrollo.** | **Este archivo se presenta como candidato a lanzamiento, a la espera de la realización de pruebas.** |
| **Avance** | **Informe de Pruebas.** | **Documento que detalla los resultados de las pruebas de calidad realizadas a la versión proporcionada, informando también sobre los cambios que deberían de hacerse.** | **Este documento presenta que el producto ha sido probado de manera exhaustiva.** |
| **Final** | **Producto Final.** | **Versión final del producto que contiene los cambios necesarios tras las pruebas de calidad, listo para su lanzamiento.** | **Este archivo presenta el resultado final del producto.** |
| **Final** | **Registro de Cambios.** | **Documento que reporta los cambios realizados en cada actualización como parte de la fase de mantenimiento, detallando las correcciones de errores y mejoras.** | **Este documento presenta cómo se ha gestionado el mantenimiento del producto tras su lanzamiento.** |

|  |
| --- |
| **7. Plan de Trabajo** |
| En la siguiente tabla define la planificación de tu Proyecto APT de acuerdo a lo requerido. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Plan de Trabajo Proyecto APT** | | | | | | |
| Competencia o unidades de competencias | Nombre de Actividades / Tareas | Descripción Actividades / Tareas | Recursos | Duración de la actividad | Responsable | Observaciones |
| **Gestión de Requerimientos** | **Toma de Requerimientos** | **Definir los requisitos funcionales y no funcionales a partir de los objetivos específicos.** | **Documentación y análisis.** | **2 semanas** | **Benjamín Polanco** | **Identificar todos los requisitos necesarios para el desarrollo de este proyecto.** |
| **Gestión de Proyectos Informáticos** | **Diseño de Arquitectura de Software** | **Diseñar los componentes que se encargarán de la nueva altura y la generación de terreno.** | **Herramientas de diseño** | **2 semanas** | **Benjamín Polanco** | **Asegurarse de que el diseño del componente cumpla con los requisitos.** |
| **Desarrollo de Soluciones de Software** | **Implementación de Altura** | **Desarrollo de los componentes que se encargarán de la nueva altura.** | **IntelliJ IDEA, Java, Mixins, GitHub** | **2 semanas** | **Benjamín Polanco** | **Probar funcionalidad a medida que esta se desarrolla** |
| **Desarrollo de Soluciones de Software** | **Implementación de Generación de Terreno** | **Desarrollo de los componentes que se encargarán de la nueva generación de terreno.** | **IntelliJ IDEA, Java, Mixins, GitHub** | **4 semanas** | **Benjamín Polanco** | **Probar funcionalidades a medida que estas se desarrollan.** |
| **Desarrollo de Soluciones de Software** | **Implementación de Entidades** | **Desarrollo de los componentes que se encargarán de la generación de entidades en los limites extendidos.** | **IntelliJ IDEA, Java, Mixins, GitHub** | **1 semana** | **Benjamín Polanco** | **Asegurarse de que la inteligencia artificial y navegación de las entidades funcione correctamente.** |
| **Desarrollo de Soluciones de Software** | **Implementación de Compatibilidad** | **Desarrollo de los componentes que se encargarán de la compatibilidad con otras modificaciones.** | **IntelliJ IDEA, Java, Mixins, GitHub** | **3 semanas** | **Benjamín Polanco** | **Asegurarse de que el proyecto sea compatible con múltiples modificaciones esenciales.** |
| **Gestión de Calidad** | **Pruebas de Calidad** | **Realizar pruebas exhaustivas para verificar que el proyecto cumpla con los requisitos definidos y sea compatible con otras modificaciones.** | **Minecraft** | **2 semanas** | **Benjamín Polanco** | **Realizar exhaustivas pruebas de calidad para asegurarse que el proyecto cumpla con los requerimientos definidos.** |
| **Gestión de Riesgos** | **Monitoreo y Actualizaciones.** | **Realizar un monitoreo del producto tras su lanzamiento, escuchando la retroalimentación de los usuarios y actualizando tras ser necesario.** | **Análisis, IntelliJ IDEA, Java, Mixins, GitHub** | **2 semanas** | **Benjamín Polanco** | **Realizar monitoreo activo al software durante las primeras 3 semanas de su lanzamiento para asegurar que funcione correctamente.** |

|  |
| --- |
| **8. Carta Gantt** |
| Busca un formato de Carta Gantt que te acomode y organiza en este las actividades planificadas en el punto anterior considerando el periodo asignado para el desarrollo de tu Proyecto APT. Debes mantener la temporalidad del periodo académico en el desarrollo de las tres fases que contempla la Asignatura de Portafolio de Título. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividad** | Fase 1 | | | | Fase 2 | | | | | | | | | | | Fase 3 | | |
| **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **S8** | **S9** | **S10** | **S11** | **S12** | **S13** | **S14** | **S15** | **S16** | **S17** | **S18** |
| **Toma de requerimientos** | **X** | **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Diseño de arquitectura de software** |  |  | **X** | **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Implementación de nuevo límite de altura** |  |  |  |  | **X** | **X** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Implementación de la nueva generación de terreno** |  |  |  |  |  |  | **X** | **X** | **X** | **X** |  |  |  |  |  |  |  |  |
| **Implementación de entidades** |  |  |  |  |  |  |  |  |  |  | **X** |  |  |  |  |  |  |  |
| **Implementación de compatibilidad** |  |  |  |  |  |  |  |  |  |  |  | **X** | **X** | **X** |  |  |  |  |
| **Pruebas de calidad** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **X** | **X** |  |  |
| **Monitoreo y actualizaciones** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **X** | **X** |