I. Guía Pedagógica del Módulo Programación de videojuegos

Contenido

		Pág.
I.	**Guía pedagógica**	
1.	Descripción	3
2.	Datos de identificación de la norma	4
3.	Generalidades pedagógicas	5
4.	Enfoque del módulo	13
5.	Orientaciones didácticas y estrategias de aprendizaje por unidad	15
6.	Prácticas/ejercicios/problemas/actividades	22
II.	**Guía de evaluación**	74
7.	Descripción	75
8.	Tabla de ponderación	79
9.	Materiales para el desarrollo de actividades de evaluación	80
10.	Matriz de valoración o rúbrica	81

1. Descripción

La Guía Pedagógica es un documento que integra elementos técnico-metodológicos planteados de acuerdo con los principios y lineamientos del **Modelo Académico del CONALEP** para orientar la práctica educativa del docente en el desarrollo de competencias previstas en los programas de estudio.

La finalidad que tiene esta guía es facilitar el aprendizaje de los alumnos, encauzar sus acciones y reflexiones y proporcionar situaciones en las que desarrollará las competencias. El docente debe asumir conscientemente un rol que facilite el proceso de aprendizaje, proponiendo y cuidando un encuadre que favorezca un ambiente seguro en el que los alumnos puedan aprender, tomar riesgos, equivocarse extrayendo de sus errores lecciones significativas, apoyarse mutuamente, establecer relaciones positivas y de confianza, crear relaciones significativas con adultos a quienes respetan no por su estatus como tal, sino como personas cuyo ejemplo, cercanía y apoyo emocional es valioso.

Es necesario destacar que el desarrollo de la competencia se concreta en el aula, ya que **formar con un enfoque en competencias significa crear experiencias de aprendizaje para que los alumnos adquieran la capacidad de movilizar, de forma integral, recursos que se consideran indispensables para saber resolver problemas en diversas situaciones o contextos,** e involucran las dimensiones cognitiva, afectiva y psicomotora; por ello, los programas de estudio, describen las competencias a desarrollar, entendiéndolas como la combinación integrada de conocimientos, habilidades, actitudes y valores que permiten el logro de un desempeño eficiente, autónomo, flexible y responsable del individuo en situaciones específicas y en un contexto dado. En consecuencia, la competencia implica la comprensión y transferencia de los conocimientos a situaciones de la vida real; ello exige relacionar, integrar, interpretar, inventar, aplicar y transferir los saberes a la resolución de problemas. Esto significa que **el contenido, los medios de enseñanza, las estrategias de aprendizaje, las formas de organización de la clase y la evaluación se estructuran en función de la competencia a formar**; es decir, el énfasis en la proyección curricular está en lo que los alumnos tienen que aprender, en las formas en cómo lo hacen y en su aplicación a situaciones de la vida cotidiana y profesional.

Considerando que el alumno está en el centro del proceso formativo, se busca acercarle elementos de apoyo que le muestren qué **competencias** va a desarrollar, cómo hacerlo y la forma en que se le evaluará. Es decir, mediante la guía pedagógica el alumno podrá **autogestionar su aprendizaje** a través del uso de estrategias flexibles y apropiadas que se transfieran y adopten a nuevas situaciones y contextos e ir dando seguimiento a sus avances a través de una autoevaluación constante, como base para mejorar en el logro y desarrollo de las competencias indispensables para un crecimiento académico y personal.

2. Datos de Identificación de la Norma

Título:.	
Unidad (es) de competencia laboral:	
Código:	**Nivel de competencia:**

3. Generalidades Pedagógicas

Con el propósito de difundir los criterios a considerar en la instrumentación de la presente guía entre los docentes y personal académico de planteles y Colegios Estatales, se describen **algunas consideraciones** respecto al desarrollo e intención de las competencias expresadas en los módulos correspondientes a la formación básica, propedéutica y profesional.

Los principios asociados a la **concepción constructivista del aprendizaje** mantienen una estrecha relación con los de la **educación basada en competencias**, la cual se ha concebido en el Colegio como el enfoque idóneo para orientar la formación ocupacional de los futuros profesionales técnicos y profesionales técnicos bachiller. Este enfoque constituye una de las opciones más viables para lograr la vinculación entre la educación y el sector productivo de bienes y servicios.

En los programas de estudio se proponen una serie de contenidos que se considera conveniente abordar para obtener los **Resultados de Aprendizaje establecidos**; sin embargo, se busca que este planteamiento le dé al docente la posibilidad de **desarrollarlos con mayor libertad y creatividad**.

En este sentido, se debe considerar que el papel que juegan el alumno y el docente en el marco del Modelo Académico del CONALEP tenga, entre otras, las siguientes características:

El alumno:	El docente:
❖ Mejora su capacidad para resolver problemas. ❖ Aprende a trabajar en grupo y comunica sus ideas. ❖ Aprende a buscar información y a procesarla. ❖ Construye su conocimiento. ❖ Adopta una posición crítica y autónoma. ❖ Realiza los procesos de autoevaluación y coevaluación.	❖ Organiza su formación continua a lo largo de su trayectoria profesional. ❖ Domina y estructura los saberes para facilitar experiencias de aprendizaje significativo. ❖ Planifica los procesos de enseñanza y de aprendizaje atendiendo al enfoque por competencias, y los ubica en contextos disciplinares, curriculares y sociales amplios. ❖ Lleva a la práctica procesos de enseñanza y de aprendizaje de manera efectiva, creativa e innovadora a su contexto institucional. ❖ Evalúa los procesos de enseñanza y de aprendizaje con un enfoque formativo. ❖ Construye ambientes para el aprendizaje autónomo y colaborativo. ❖ Contribuye a la generación de un ambiente que facilite el desarrollo sano e integral de los estudiantes. ❖ Participa en los proyectos de mejora continua de su escuela y apoya la gestión institucional.

En esta etapa se requiere una mejor y mayor organización académica que apoye en forma relativa la actividad del alumno, que en este caso es mucho mayor que la del docente; lo que no quiere decir que su labor sea menos importante. **El docente en lugar de transmitir vertical y unidireccionalmente los conocimientos, es un mediador del aprendizaje**, ya que:

- Planea y diseña experiencias y actividades necesarias para la adquisición de las competencias previstas. Asimismo, define los ambientes de aprendizaje, espacios y recursos adecuados para su logro.
- Proporciona oportunidades de aprendizaje a los estudiantes apoyándose en metodologías y estrategias didácticas pertinentes a los Resultados de Aprendizaje.
- Ayuda también al alumno a asumir un rol más comprometido con su propio proceso, invitándole a tomar decisiones.
- Facilita el aprender a pensar, fomentando un nivel más profundo de conocimiento.
- Ayuda en la creación y desarrollo de grupos colaborativos entre los alumnos.
- Guía permanentemente a los alumnos.
- Motiva al alumno a poner en práctica sus ideas, animándole en sus exploraciones y proyectos.

Considerando la importancia de que el docente planee y despliegue con libertad su experiencia y creatividad para el desarrollo de las competencias consideradas en los programas de estudio y especificadas en los Resultados de Aprendizaje, en las competencias de las Unidades de Aprendizaje, así como en la competencia del módulo; **podrá proponer y utilizar todas las estrategias didácticas que considere necesarias** para el logro de estos fines educativos, con la recomendación de que fomente, preferentemente, las estrategias y técnicas didácticas que se describen en este apartado.

Al respecto, entenderemos como estrategias didácticas los planes y actividades orientados a un desempeño exitoso de los resultados de aprendizaje, que incluyen estrategias de enseñanza, estrategias de aprendizaje, métodos y técnicas didácticas, así como, acciones paralelas o alternativas que el docente y los alumnos realizarán para obtener y verificar el logro de la competencia; bajo este tenor, **la autoevaluación debe ser considerada también como una estrategia por excelencia para educar al alumno en la responsabilidad y para que aprenda a valorar, criticar y reflexionar sobre el proceso de enseñanza y su aprendizaje individual**.

Es así como la selección de estas estrategias debe orientarse hacia un enfoque constructivista del conocimiento y estar dirigidas a que **los alumnos observen y estudien su entorno**, con el fin de generar nuevos conocimientos en contextos reales y el desarrollo de las capacidades reflexivas y críticas de los alumnos.

Desde esta perspectiva, a continuación se describen brevemente los tipos de aprendizaje que guiarán el diseño de las estrategias y las técnicas que deberán emplearse para el desarrollo de las mismas:

TIPOS DE APRENDIZAJES.

Significativo

Se fundamenta en una concepción constructivista del aprendizaje, la cual se nutre de diversas concepciones asociadas al cognoscitivismo, como la teoría psicogenética de Jean Piaget, el enfoque sociocultural de Vygotsky y la teoría del aprendizaje significativo de Ausubel.

Dicha concepción sostiene que el ser humano tiene la disposición de **aprender verdaderamente sólo aquello a lo que le encuentra sentido** en virtud de que está vinculado con su entorno o con sus conocimientos previos. Con respecto al comportamiento del alumno, se espera que sean capaces de desarrollar aprendizajes significativos, en una amplia gama de situaciones y circunstancias, lo cual equivale a **"aprender a aprender"**, ya que de ello depende la construcción del conocimiento.

Colaborativo.

El aprendizaje colaborativo puede definirse como el conjunto de métodos de instrucción o entrenamiento para uso en grupos, así como de estrategias para propiciar el desarrollo de habilidades mixtas (aprendizaje y desarrollo personal y social). En el aprendizaje colaborativo **cada miembro del grupo es responsable de su propio aprendizaje, así como del de los restantes miembros del grupo** (Johnson, 1993.)

Más que una técnica, el aprendizaje colaborativo es considerado una filosofía de interacción y una forma personal de trabajo, que implica el manejo de aspectos tales como el **respeto a las contribuciones y capacidades individuales de los miembros del grupo** (Maldonado Pérez, 2007). Lo que lo distingue de otro tipo de situaciones grupales, es el desarrollo de la interdependencia positiva entre los alumnos, es decir, de una toma de conciencia de que **sólo es posible lograr las metas individuales de aprendizaje si los demás compañeros del grupo también logran las suyas**.

El aprendizaje colaborativo surge a través de transacciones entre los alumnos, o entre el docente y los alumnos, en un proceso en el cual cambia la responsabilidad del aprendizaje, del docente como experto, al alumno, y asume que el docente es también un sujeto que aprende. Lo más importante en la formación de grupos de trabajo colaborativo es vigilar que los elementos básicos estén claramente estructurados en cada sesión de trabajo. Sólo de esta manera se puede lograr que se produzca, tanto el esfuerzo colaborativo en el grupo, como una estrecha relación entre la colaboración y los resultados (Johnson & F. Johnson, 1997).

Los elementos básicos que deben estar presentes en los grupos de trabajo colaborativo para que éste sea efectivo son:

- la interdependencia positiva.
- la responsabilidad individual.
- la interacción promotora.
- el uso apropiado de destrezas sociales.
- el procesamiento del grupo.

Asimismo, el trabajo colaborativo se caracteriza principalmente por lo siguiente:

- Se desarrolla mediante **acciones de cooperación, responsabilidad, respeto y comunicación**, en forma sistemática, entre los integrantes del grupo y subgrupos.
- Va **más allá que sólo el simple trabajo en equipo** por parte de los alumnos. Básicamente se puede orientar a que los alumnos intercambien información y trabajen en tareas hasta que todos sus miembros las han entendido y terminado, aprendiendo a través de la colaboración.
- Se distingue por el desarrollo de una **interdependencia positiva entre los alumnos**, en donde se tome conciencia de que sólo es posible lograr las metas individuales de aprendizaje si los demás compañeros del grupo también logran las suyas.
- Aunque en esencia esta estrategia promueve la actividad en pequeños grupos de trabajo, se debe cuidar en el planteamiento de las actividades que **cada integrante obtenga una evidencia personal para poder integrarla a su portafolio de evidencias**.

Aprendizaje Basado en Problemas.

Consiste en la presentación de **situaciones reales o simuladas** que requieren la aplicación del conocimiento, en las cuales el **alumno debe analizar la situación y elegir o construir una o varias alternativas para su solución** (Díaz Barriga Arceo, 2003). Es importante aplicar esta estrategia ya que **las competencias se adquieren en el proceso de solución de problemas** y en este sentido, el alumno aprende a solucionarlos cuando se enfrenta a problemas de su vida cotidiana, a problemas vinculados con sus vivencias dentro del Colegio o con la profesión. Asimismo, el alumno se apropia de los conocimientos, habilidades y normas de comportamiento que le permiten la aplicación creativa a nuevas situaciones sociales, profesionales o de aprendizaje, por lo que:

- Se puede trabajar en forma individual o de grupos pequeños de alumnos que se reúnen a analizar y a resolver un problema seleccionado o diseñado especialmente para el logro de ciertos resultados de aprendizaje.
- Se debe presentar primero el problema, se identifican las necesidades de aprendizaje, se busca la información necesaria y finalmente se regresa al problema con una solución o se identifican problemas nuevos y se repite el ciclo.

- Los problemas deben estar diseñados para motivar la búsqueda independiente de la información a través de todos los medios disponibles para el alumno y además generar discusión o controversia en el grupo.
- El mismo diseño del problema debe estimular que los alumnos utilicen los aprendizajes previamente adquiridos.
- El diseño del problema debe comprometer el interés de los alumnos para examinar de manera profunda los conceptos y objetivos que se quieren aprender.
- El problema debe estar en relación con los objetivos del programa de estudio y con problemas o situaciones de la vida diaria para que los alumnos encuentren mayor sentido en el trabajo que realizan.
- Los problemas deben llevar a los alumnos a tomar decisiones o hacer juicios basados en hechos, información lógica y fundamentada, y obligarlos a justificar sus decisiones y razonamientos.
- Se debe centrar en el alumno y no en el docente.

TÉCNICAS

Método de proyectos.

Es una técnica didáctica que incluye actividades que pueden requerir que los alumnos **investiguen, construyan y analicen información** que coincida con los objetivos específicos de una tarea determinada en la que se **organizan actividades desde una perspectiva experiencial**, donde el alumno aprende a través de la práctica personal, activa y directa con el propósito de aclarar, reforzar y construir aprendizajes (Intel Educación).

Para definir proyectos efectivos se debe considerar principalmente que:

- Los alumnos son el centro del proceso de aprendizaje.
- Los proyectos se enfocan en resultados de aprendizaje acordes con los programas de estudio.
- Las preguntas orientadoras conducen la ejecución de los proyectos.
- Los proyectos involucran múltiples tipos de evaluaciones continuas.
- El proyecto tiene conexiones con el mundo real.
- Los alumnos demuestran conocimiento a través de un producto o desempeño.
- La tecnología apoya y mejora el aprendizaje de los alumnos.
- Las destrezas de pensamiento son integrales al proyecto.

Para el presente módulo se hacen las siguientes recomendaciones:

- Integrar varios módulos mediante el método de proyectos, lo cual es ideal para desarrollar un trabajo colaborativo.
- En el planteamiento del proyecto, cuidar los siguientes aspectos:
    - ✓ Establecer el alcance y la complejidad.
    - ✓ Determinar las metas.
    - ✓ Definir la duración.
    - ✓ Determinar los recursos y apoyos.
    - ✓ Establecer preguntas guía. Las preguntas guía conducen a los alumnos hacia el logro de los objetivos del proyecto. La cantidad de preguntas guía es proporcional a la complejidad del proyecto.
    - ✓ Calendarizar y organizar las actividades y productos preliminares y definitivos necesarias para dar cumplimiento al proyecto.
- Las actividades deben ayudar a responsabilizar a los alumnos de su propio aprendizaje y a **aplicar competencias adquiridas** en el salón de clase **en proyectos reales**, cuyo planteamiento se basa en un problema real e **involucra distintas áreas**.
- El proyecto debe implicar que los alumnos **participen en un proceso de investigación**, en el que **utilicen diferentes estrategias de estudio**; puedan participar en el proceso de planificación del propio aprendizaje y les ayude a ser flexibles, reconocer al "otro" y comprender su propio entorno personal y cultural. Así entonces se debe favorecer el desarrollo de **estrategias de indagación, interpretación y presentación del proceso seguido**.
- De acuerdo a algunos teóricos, mediante el método de proyectos los alumnos buscan soluciones a problemas no convencionales, cuando llevan a la práctica el hacer y depurar preguntas, debatir ideas, hacer predicciones, diseñar planes y/o experimentos, recolectar y analizar datos, establecer conclusiones, comunicar sus ideas y descubrimientos a otros, hacer nuevas preguntas, crear artefactos o propuestas muy concretas de orden social, científico, ambiental, etc.
- En la gran mayoría de los casos los proyectos se llevan a cabo **fuera del salón de clase** y, dependiendo de la orientación del proyecto, en muchos de los casos pueden **interactuar con sus comunidades** o permitirle un **contacto directo con las fuentes de información** necesarias para el planteamiento de su trabajo. Estas experiencias en las que se ven involucrados hacen que aprendan a manejar y usar los recursos de los que disponen como el tiempo y los materiales.
- Como medio de evaluación se recomienda que todos los proyectos tengan **una o más presentaciones del avance para evaluar resultados** relacionados con el proyecto.
- Para conocer acerca del progreso de un proyecto se puede:
    - ✓ Pedir reportes del progreso.
    - ✓ Presentaciones de avance,

- ✓ Monitorear el trabajo individual o en grupos.
- ✓ Solicitar una bitácora en relación con cada proyecto.
- ✓ Calendarizar sesiones semanales de reflexión sobre avances en función de la revisión del plan de proyecto.

Estudio de casos.

El estudio de casos es una técnica de enseñanza en la que los alumnos **aprenden sobre la base de experiencias y situaciones de la vida real**, y se permiten así, construir su propio aprendizaje en un contexto que los aproxima a su entorno. Esta técnica se basa en la participación activa y en procesos colaborativos y democráticos de discusión de la situación reflejada en el caso, por lo que:

- Se deben representar situaciones problemáticas diversas de la vida para que se estudien y analicen.
- Se pretende que los alumnos generen soluciones validas para los posibles problemas de carácter complejo que se presenten en la realidad futura.
- Se deben proponer datos concretos para reflexionar, analizar y discutir en grupo y encontrar posibles alternativas para la solución del problema planteado. Guiar al alumno en la generación de alternativas de solución, le permite desarrollar la habilidad creativa, la capacidad de innovación y representa un recurso para conectar la teoría a la práctica real.
- Debe permitir reflexionar y contrastar las propias conclusiones con las de otros, aceptarlas y expresar sugerencias.

El estudio de casos es pertinente usarlo cuando se pretende:

- Analizar un problema.
- Determinar un método de análisis.
- Adquirir agilidad en determinar alternativas o cursos de acción.
- Tomar decisiones.

Algunos teóricos plantean las siguientes fases para el estudio de un caso:

- **Fase preliminar:** Presentación del caso a los participantes
- **Fase de eclosión:** "Explosión" de opiniones, impresiones, juicios, posibles alternativas, etc., por parte de los participantes.
- **Fase de análisis:** En esta fase es preciso llegar hasta la determinación de aquellos hechos que son significativos. Se concluye esta fase cuando se ha conseguido una síntesis aceptada por todos los miembros del grupo.

- **Fase de conceptualización:** Es la formulación de conceptos o de principios concretos de acción, aplicables en el caso actual y que permiten ser utilizados o transferidos en una situación parecida.

Interrogación.

Consiste en llevar a los alumnos a la **discusión y al análisis de situaciones o información**, con base en preguntas planteadas y formuladas por el docente o por los mismos alumnos, con el fin de explorar las capacidades del pensamiento al activar sus procesos cognitivos; se recomienda **integrar esta técnica de manera sistemática y continua** a las anteriormente descritas y al abordar cualquier tema del programa de estudio.

Participativo-vivenciales.

Son un conjunto de elementos didácticos, sobre todo los que exigen un grado considerable de **involucramiento y participación de todos los miembros del grupo** y que sólo tienen como límite el grado de imaginación y creatividad del facilitador.

Los ejercicios vivenciales son una alternativa para llevar a cabo el proceso enseñanza-aprendizaje, no sólo porque facilitan la transmisión de conocimientos, sino porque además permiten **identificar y fomentar aspectos de liderazgo, motivación, interacción y comunicación del grupo**, etc., los cuales son de vital importancia para la organización, desarrollo y control de un grupo de aprendizaje.

Los ejercicios vivenciales resultan ser una situación planeada y estructurada de tal manera que representan una experiencia muy atractiva, divertida y hasta emocionante. El juego significa apartarse, salirse de lo rutinario y monótono, para asumir un papel o personaje a través del cual el individuo pueda manifestar lo que verdaderamente es o quisiera ser sin temor a la crítica, al rechazo o al ridículo.

El desarrollo de estas experiencias se encuentra determinado por los conocimientos, habilidades y actitudes que el grupo requiera revisar o analizar y por sus propias vivencias y necesidades personales.

4. Enfoque del Módulo

El módulo de **Programación de videojuegos** se ha elaborado con el propósito de desarrollar en el alumno las competencias necesarias en el entorno relativo al Trayecto de programación multimedia, en específico de desarrollo de software de entretenimiento; situación que hace necesario tomar como punto de partida la planeación de un proyecto de software. Es necesario ubicar el contexto tecnológico, plataformas, dispositivos y entorno para la construcción de videojuegos que brinden una solución en ámbitos donde se requieran productos de entretenimiento, comunicación y aprendizaje; aprovechando la diversidad de tecnologías orientadas para la producción de aplicaciones de este tipo.

El módulo pretende el desarrollo de proyectos en programación de videojuegos básicos, realización de tareas aplicadas para el logro de un producto final, que plantea como imprescindible que los estudiantes observen, deduzcan y planteen hipótesis sobre los contenidos que trabajen y las experiencias que vivan con el logro de los proyectos, favoreciendo así la reflexión acerca de las posibilidades de aplicación de lo que aprendan a su vida diaria, desarrollar una visión prospectiva donde recurra a lo que sabe para enfrentar situaciones desconocidas, construir fórmulas personales para solucionar problemas, desarrollar la capacidad de organizar actividades escolares y colectivas, encontrar las propias estrategias de aprendizaje y finalmente asumir compromisos para el logro de metas.

En este módulo se desarrollan las siguientes unidades. La primera unidad aborda lo referente a la determinación de características, tecnologías de software y un plan de acción que permitan el modelado y diseño de videojuegos, y que como base sustenta a la segunda unidad, que busca el dominio y uso del entorno de programación y ayuda de herramientas necesarias para la construcción e implementación de videojuegos. Para llegar a este propósito es aconsejable que el módulo sea abordado con planteamiento de problemas de la vida cotidiana y de interés en el ámbito académico y profesional que lleven al estudiante al análisis de la situación y a construir o tomar alternativas para su solución.

El contexto en el que se estructura el módulo parte de la consideración de que en la sociedad actual se exige que la escuela “forme” y “prepare para la vida”, de modo que el alumno no sólo responda a situaciones inmediatas sino que adquiera conocimientos, habilidades y actitudes que le permitan una participación reflexiva y comprometida con su entorno local y mundial. Para lograrlo hay que enfatizar el desarrollo de competencias que permitan a los sujetos comprender el mundo e insertarse de manera exitosa a la sociedad,

El desarrollo de estas competencias-implica interrelaciones mutuas. Así, el promover la cultura del respeto y la solidaridad se vincula fácilmente con la capacidad de tomar decisiones y la iniciativa de llevar a cabo proyectos personales, aprovechando diversas informaciones y los avances de la ciencia. El reto docente es cómo integrar las competencias y favorecer el desarrollo de las disciplinas que pretenda enseñar ya que el enfoque de competencias

que se plantea no alude únicamente a las “competencias para la vida”, sino a las competencias básicas, disciplinares y profesionales que forman un todo.

Para fomentar el desarrollo de las capacidades mencionadas, el docente debe considerar las competencias ya adquiridas de los alumnos en los módulos precedentes de la carrera y en específico en este trayecto técnico, a fin de que ello lo motive a adquirir nuevos conocimientos y experiencias que se integren de forma significativa a las estructuras cognitivas que ya posee, ya sea a través de lo que él mismo descubra o infiera, o a través del análisis y síntesis creativa de los planteamientos docentes. En lo que se refiere al aprendizaje procedimental, implica la consecución del propósito del módulo a través de acciones secuenciadas que lleven gradualmente al alumno al desarrollo de sus actividades, primeramente académicas y posteriormente profesionales, de manera segura, consciente y responsable. Por otra parte, es importante incluir y promover en este módulo estrategias de aprendizaje colaborativo y grupal, así como fomentar el desarrollo de competencias transversales que permitan establecer una mejor comunicación e interrelaciones con los demás, socializar, compartir e intercambiar información, potencializar un pensamiento crítico, lo que contribuye a activar el aprendizaje y autoaprendizaje.

Se recomienda por último elaborar un código ético durante el desarrollo del módulo con el propósito de definir los compromisos y responsabilidades que deben compartir en el espacio académico, como: respeto a la persona, honestidad, confianza, justicia, comunicación, cooperación, iniciativa, amabilidad, perseverancia y la actitud positiva para el logro de objetivos, así como adecuar las prácticas de ejercicio al equipo existente en el laboratorio de informática y al área de aplicación de la carrera; realizando las prácticas con orden, limpieza, fomentando el uso de software libre o de marca, evitando acciones ilegales para garantizar el funcionamiento y calidad del mismo.

5. Orientaciones didácticas y estrategias de aprendizaje por unidad

Unidad I	**Diseño de videojuegos**

Orientaciones Didácticas

La unidad correspondiente al **Diseño de videojuegos**, se avoca en el plan de diseño, uso de tecnología y herramientas de software técnicas necesarias que sustenten la base para el desarrollo del proyecto de software y la planeación de videojuegos. El desarrollo de esta unidad proporcionará al alumno elementos básicos que le permitirán desarrollar las actividades y prácticas propias de esta competencia y apoye a la unidad subsecuente, por eso se propone que el docente lleve a cabo lo siguiente:

- Analiza con sus alumnos, las implicaciones y alcances del programa del módulo, a través de dinámica grupal de encuadre, con el fin de precisar formas de trabajar, responsabilidades y compromisos de los integrantes del grupo que dirijan al logro tanto del propósito del módulo, como del objetivo de este trayecto de la carrera.
- Aplica una prueba exploratoria sobre técnicas de programación, manejo de lenguajes de programación, teoría de general sobre desarrollo de aplicaciones de software y bases de datos como evaluación diagnóstica.
- Considera que el resultado de aprendizaje de determinar las características para el desarrollo de videojuegos con base en tecnologías y herramientas de software, se encuentra estrechamente vinculado con el subsecuente resultado de aprendizaje sobre planear el diseño de videojuegos mediante metodología específica, tareas de ejecución y ayuda de herramientas de software, y para lograrlo se sugiere que el docente opere con los conceptos y habilidades para al diseño de videojuegos.
- Analiza la información relacionada con el manejo del entorno de desarrollo de videojuegos, precisando la utilidad de esta fase de diseño, clasificaciones, tendencias y tecnologías existentes resaltan en el mercado laboral y en el campo de TI sobre diseño de videojuegos
- Fomenta la búsqueda de información e investigaciones para dar respuesta a preguntas específicas referentes a identificación de elementos de videojuego, herramientas de software, conocimientos base de programación y distinción de dispositivos.
- Demuestra el desglose del diseño de videojuegos, en cuanto planteamiento y aterrizaje de la idea a desarrollar y la preparación de las herramientas necesarias.
- Conduce y apoya a los alumnos en la planeación del diseño de videojuegos por medio de elaboración de mapas conceptuales, cuadros sinópticos, simulaciones, estudios de caso sobre la metodología específica, tareas de ejecución y ayuda de herramientas de software.
- Alienta la participación de los alumnos como facilitador o moderador en la discusión grupal para la realización de ejercicios prácticos, exposiciones, demostraciones, simulaciones, prácticas de ejercicio, comentarios, conclusiones, recapitulaciones y coevaluación.

Fortalece las siguientes competencias transversales

- Propone maneras de solucionar un problema o desarrollar un proyecto en equipo, definiendo un curso de acción con pasos específicos que conlleven a la formulación de una propuesta de solución para diseño de videojuegos.
- Define metas y da seguimiento a sus procesos de construcción de conocimiento para la implementación de la propuesta de solución planteada.
- Promueve una dinámica grupal colaborativa y cooperativa durante el transcurso de cada sesión para favorecer un clima que fomente el intercambio constructivo de ideas.
- Subraya la importancia que tiene la presencia del alumno en cada clase, su participación para el enriquecimiento del aprendizaje de todo el grupo y la asignación de tareas y actividades intra y extramuros, con el fin de incentivar en él su cumplimiento voluntario y oportuno.
- Fomenta actividades de auto-estudio y de autocrítica de los alumnos con la evaluación de sus propios juicios.
- Remover las visiones que obstaculizan el libre acceso a opciones de formación y de trabajo; fomentando **la igualdad de oportunidades**, la participación democrática, la multiculturalidad, la atención a grupos desfavorecidos y su inserción social, laboral y económica, así como el **respeto a la persona sin discriminación por su sexo, origen, situación social, conocimientos, etc.**

Estrategias de Aprendizaje	Recursos Académicos
• Realizar una investigación documental a través de la consulta de las fuentes sugeridas o de páginas en Internet sobre el entorno de desarrollo de videojuegos y sus tendencias. • Elaborar un cuadro comparativo y de caracterización de la investigación realizada • Organizar equipos de trabajo con la totalidad de integrantes del grupo, para analizar los siguientes temas y ejemplificar ante el grupo las características de su utilización: de las distintas tecnologías para desarrollo de videojuegos mediante la ubicación del mercado actual; diferenciación de plataformas o tecnologías, consolas y sus características; así como los beneficios de los productos de videojuegos y sus aplicaciones. • Exponer de manera grupal un mapa conceptual sobre el entorno de diseño de videojuegos para su retroalimentación y/ o corrección. • Plantear de manera individual esquemas de diferentes tipos de diseños de soluciones de videojuegos y definir los objetivos del problema. • **Realizar la actividad No. 1** “Diferenciación de tipos de videojuegos, consolas y características”. • Compilar la información de la actividad anterior y presentar cuadro comparativo de investigación, elaborado en un documento de texto en formato impreso y/o digital. • **Realizar la actividad de evaluación 1.1.1** “Realiza un cuadro comparativo donde se consideren los elementos dispositivos y herramientas para el desarrollo de videojuegos”. • Comentar en clase los resultados de la actividad de evaluación realizada, efectuando una coevaluación enfocada tanto al proceso ejecutado como a los resultados obtenidos. • Buscar fuentes de información en base páginas web sugeridas y plasmar en un mapa cognitivo	**Básica:** • Ceballos, Fco. Javier. **Java 2 - Curso De Programación** - 4a. Edición, México, Alfaomega Ra-Ma, 2006. • Ceballos, Fco. Javier. **Microsoft C# - Curso De Programación**. México, Editorial Alfaomega, 2008. • Ramírez, Felipe. **Introducción a la programación, algoritmos y su implementación en VB.NET, C#, Java, C++.** 2da. Edición, México, Alfaomega Grupo Editor, 2007. **Complementaria:** • López Román, Leobardo. **Metodología de la programación orientada a objetos**, 1ª. Edición, México, Editorial Alfaomega, 2006. • Marshal, Donis. **Programming Microsoft Visual C# 2008**: **The Language. Microsoft**. Estados Unidos, Mayo de 2008. • López Román, Leobardo, **Programación Estructurada y Orientada A Objetos** 3ª.

Estrategias de Aprendizaje	Recursos Académicos

el panorama del diseño para videojuegos considerando los elementos de:
- Tipo de juego a hacer
- Elementos de software necesarios:
  – Utilización de dibujos, bosquejos, story board de las pantallas.
  – Lenguajes de desarrollo –(*C, C++, Java*)
  – Herramientas de desarrollo para crear las imágenes (*Fractal Design Painter, 3D Studio, Paintbrush, Photoshop, Corel Draw*).
- Disposición de equipo de cómputo.
- Equipo de desarrollo (programadores, dibujantes. técnicos en sonido)
- Generación del plan de acción o guía de producción
• Elaborar un cuadro sinóptico por equipos y discutir en grupo sobre los elementos para el desarrollo de diseño de videojuegos.
• Identifica en dúos los elementos de un Plan de implementación como sigue.
- Definición del objetivo de la aplicación de videojuego a desarrollar.
- Tecnología a usar en el desarrollo de la aplicación.
- Diseño
- Elaboración
- Pruebas.
- Estimación de costos.
• **Realizar la actividad No. 2** "Propuesta del diseño de videojuego".
• Compilar la información de la actividad anterior y presentar propuesta de solución de videojuego con su planeación, elaborado en un documento de texto en formato impreso y/o digital.
• **Realizar la actividad de evaluación 1.2.1 "**Elabora un plan de diseño y acción para videojuegos con la siguiente estructura":
• Diseño del proyecto.
– Creación de historia
– Objetivos
– Recursos necesarios
– Arte conceptual
– Sonido
– Mecánica de juego
– Diseño de programación
• Cronograma de actividades
• Comentar en clase los resultados de la actividad de evaluación realizada, efectuando una coevaluación enfocada tanto al proceso ejecutado como a los resultados obtenidos.
	Edición, México, Editorial Alfaomega, 2011.
	• ARCE, Francisco Javier. **ActionScript 3.0 - Aprenda a programar**. México, Alfaomega, 2011.
	• MEDIAactive. **APRENDER 3DS MAX 2010. Con 100 ejercicios prácticos** - **Aprenda a programar.** México, Alfaomega, 2011
	
	Páginas Web:
	• ¿Cómo empezar en el Desarrollo de Videojuegos?, **Disponible en: http://www.losersjuegos.com.ar/referencia/articulos/como_empezar (19-08-2015)**.
	• Introducción a la programación de juegos - Nacho Cabanes, **Disponible en: http://www.nachocabanes.com/videojuegos/ipj/index.php (19-08-2015)**.
	• Crea videojuegos, curso de diseño y programación de videojuegos, **Disponible en: http://www.creavideojuegos.com/index.html (19-08-2015)**.

Unidad II	**Construcción de videojuegos**

Orientaciones Didácticas

La unidad correspondiente a la **Construcción de videojuegos**, se avoca a que el alumno este en posibilidades de desarrollar aplicaciones de videojuegos simples o complejos considerando los elementos del entorno de programación, lenguaje determinado, y ayuda de herramientas que conlleven a la generación de un producto de software de entretenimiento. El desarrollo de esta unidad proporcionará al alumno elementos necesarios para desarrollar las actividades y prácticas propias de esta competencia, por eso se propone que el docente lleve a cabo lo siguiente:

- Considera que el resultado de aprendizaje del manejo de los elementos del entorno de desarrollo para videojuegos mediante los componentes y herramientas de lenguajes programación, se encuentra estrechamente vinculado con el subsecuente resultado de aprendizaje sobre Programa aplicaciones de videojuego de acuerdo al diseño establecido y a la codificación en lenguaje de programación específico; y para lograrlo se sugiere que el docente opere con los conceptos y habilidades construidos conjuntamente con sus alumnos en lo que se refiere al desarrollo de esta unidad de aprendizaje para proceder a la construcción propia del videojuego.
- Organiza sistemáticamente la información que se ha de manejar y procesar para su aprendizaje. Efectuando explícitamente la vinculación de esta unidad con la que precede.
- Analiza la información relacionada con la fase de construcción de videojuegos, precisando la importancia y tecnologías existentes, lenguajes y plataformas de programación sobre el desarrollo de videojuegos.
- Propone planteamientos de problemas y su solución, mediante la elaboración de ejercicios de análisis de casos reales aplicados.
- Demuestra el procedimiento de elaboración de una aplicación de videojuegos mediante herramientas de programación y presenta prototipos, modelos o estudios de caso; así como técnicas de diseño de programas, estructuras de datos y configuración.
- Promueve el uso de las herramientas de programación de lenguaje C, C++ o Java a través del el manejo de sus componentes para la construcción de las aplicaciones desarrolladas por los alumnos.
- Efectúa el proceso de evaluación continua que haga referencia al proceso sistemático y permanente mediante el cual se haya valorado el logro de los objetivos planteados y el desarrollo de resolución de problemas por parte del alumno.
- Realiza el cierre de ciclos de aprendizaje no solamente al concluir cada tema o subtema, sino de cada sesión de clase, con la finalidad de lograr un proceso lógico de enseñanza-aprendizaje, en el que el alumno pueda apreciar tanto sus logros cotidianos y la importancia de su esfuerzo y constancia, como la importancia de la afirmación de sus capacidades para dar paso a la adquisición de nuevas competencias, especialmente las relacionadas con el manejo de tecnologías de información y la comunicación para procesar u obtener datos, así como expresar ideas.

Fortalece las siguientes competencias transversales

- Ordenar información de acuerdo a categorías, jerarquías y relaciones que permitan elaborar aplicaciones de videojuegos.
- Utiliza las TICs para procesar e interpretar información y desarrollar aplicaciones de videojuegos.
- Fomenta la búsqueda de información para dar respuesta a preguntas específicas referentes a forma de construir y programar aplicaciones de videojuegos, a fin de lograr generar un producto de entretenimiento.
- Brindar una formación de calidad y con equidad en donde se promueva la **participación plena de los sujetos** en el mundo del trabajo, el estudio y la convivencia acompañando sus procesos de reconocimiento y adquisición de saberes y habilidades, procurando **remover inequidades** que se originan en visiones estereotipadas sobre el papel que juegan las distintas personas según su sexo, origen, situación social, conocimientos, etc.

Estrategias de Aprendizaje	Recursos Académicos
• Investigar información específica y discutir sobre la fase de construcción de videojuegos, después de un diseño establecido que responda a lo siguiente: - Concepto de plataforma .NET - Estructura de una solución. - Estructura de un proyecto. - Codificación de rutinas estructuradas de programación - Codificación de rutinas y tareas en en programación orientada a objetos - Compiladores de lenguejes C, C++, Java - Aplicación de pruebas de funcionamiento de videojuegos. • Analizar en grupos los factores que impactan a la construcción de la aplicación de videojuego. • Construir rutinas de programación de aplicaciones para videojuegos en grupos de trabajo mediante las herramientas de lenguajes de programación programación considerando: - Codificación de rutinas estructuradas de programación - Codificación de rutinas y tareas en en programación orientada a objetos - Compiladores de lenguejes C, C++, Java - Ejecución de la aplicación de videojuego realizada. • **Realizar la práctica No. 1** “Uso del modo gráfico y dibujando con herramientas del lenguaje”. • **Realizar la práctica No. 2** “Leer del teclado y escribir texto usando lenguaje de programación”. • Presentar código fuente generado, en formato impreso y/o digital, haciendo acopio de la información derivada de las prácticas anteriores. • **Realizar la actividad de evaluación 2.1.1** “Realiza rutinas de programación estructurada o POO aplicado a casos prácticos con lenguaje de programación seleccionado donde se genere el código fuente y código ejecutable”.	**Básica:** • Ceballos, Fco. Javier. **Java 2 - Curso De Programación** - 4a. Edición, México, Alfaomega Ra-Ma, 2006. • Ceballos, Fco. Javier. **Microsoft C# - Curso De Programación**. México, Editorial Alfaomega, 2008. • Ramírez, Felipe. **Introducción a la programación, algoritmos y su implementación en VB.NET, C#, Java, C++.** 2da. Edición, México, Alfaomega Grupo Editor, 2007. **Complementaria:** • **Microsoft Visual C# 2008**: The Language. Microsoft. Estados Unidos. Mayo de 2008 • MEDIAactive. **APRENDER 3DS MAX 2010. Con 100 ejercicios prácticos** - Aprenda a programar. México, Alfaomega, 2011

Estrategias de Aprendizaje	Recursos Académicos
• Comentar en clase los resultados de la actividad de evaluación realizada, efectuando una coevaluación enfocada tanto al proceso ejecutado como a los resultados obtenidos. • **Realizar la práctica No. 3** "Generar números al azar con método al aleatorio". • **Realizar la práctica No.4** "Elaborar el juego del ahorcado con método al aleatorio". • **Realizar la práctica No. 5** "Elaborar juego evitando esperar al teclado, motos de luz". • **Realizar la práctica No. 6** "Crea Juego Mini Serpiente 1 con Uso de Mapas". • **Realizar la práctica No. 7** "Crear juego Miniserpiente 2 usando lenguaje de programación". • **Realizar la práctica No. 8** "Evitar los parpadeos. Juego Miniserpiente 3 mediante aproximaciones". • **Realizar la práctica No. 9** "Crear juego usando la paleta de colores". • **Realizar la práctica No. 10** "Crear juego usando funciones del mouse". • **Realizar la práctica No. 11** "Aplicación de matemáticas para juegos. Juego: TiroAlPlato usando lenguaje de programación". • **Realizar la práctica No. 12** "Reproducir sonidos. Juego: SimeonDice usando lenguaje de programación". • **Realizar la práctica No. 13** "Uso de formatos de archivos de imágenes usando lenguaje de programación". • **Realizar la práctica No. 14** "Movimientos de acercamiento usando lenguaje de programación". • **Realizar la práctica No. 15** ""Manejo de distintas aproximaciones usando lenguaje de programación" • Compilar la información de las prácticas anteriores y presentar código fuente generado, en formato impreso y/o digital. • Desarrollar los siguientes ejercicios de construcción de programas de videojuegos en forma libre y extracurricular mediante herramientas de lenguajes de programación y herramientas de TI, basándose en las siguientes referencias de Internet: ✓ "Avanzando Columnas" http://www.nachocabanes.com/videojuegos/ipj/ipj26.php ✓ "La aproximación orientada a objetos. Toma de contacto con un primer "arcade": MiniMiner 1". http://www.nachocabanes.com/videojuegos/ipj/ipj28.php ✓ "La aproximación orientada a objetos. MiniMiner 2: Aislando del hardware". .http://www.nachocabanes.com/videojuegos/ipj/ipj29.php ✓ "La aproximación orientada a objetos. MiniMiner 3: Personaje y enemigo como clases".	**Páginas Web:** • Introducción a la programación de juegos - Nacho Cabanes, **Disponible en: http://www.nachocabanes.com/videojuegos/ipj/index.php (19-08-2015)**. • Curso de iniciación a la programación con C#, **Disponible en: http://www.elguille.info/NET/cursoCSharpErik/index.htm (19-08-2015)**. • Tutorial pygame 1: introducción a la programacion de videojuegos, **Disponible en: http://pythonmania.wordpress.com/2010/03/23/tutorial-pygame-introduccion/ (19-08-2015)**. • Curso de iniciación a la programación de juegos. **Disponible en: http://www.publispain.com/supertutoriales/programacion/videojuegos/cursos/1/1div_3.htm (19-08-2015)**. • Crea videojuegos, curso de diseño y programación de videojuegos, **Disponible en: http://www.creavideojuegos.com/index.html (19-08-2015)**.

Estrategias de Aprendizaje	Recursos Académicos
http://www.nachocabanes.com/videojuegos/ipj/ipj30.php ✓ "La aproximación orientada a objeto. MiniMiner 4: Una pantalla de juego real". http://www.nachocabanes.com/videojuegos/ipj/ipj31.php ✓ "Colisiones con enemigos. Perder vidas. Aplicación a MiniMiner 5". http://www.nachocabanes.com/videojuegos/ipj/ipj32.php ✓ "Volver a comenzar una partida. Una pantalla de presentación animada. Aplicación a MiniMiner 6". http://www.nachocabanes.com/videojuegos/ipj/ipj33.php ✓ "Añadiendo funcionalidades a "MiniMiner" (1): chocar con el fondo, saltar. http://www.nachocabanes.com/videojuegos/ipj/ipj34.php ✓ "Una consola para depuración". http://www.nachocabanes.com/videojuegos/ipj/ipj35.php ✓ "Añadiendo funcionalidades a "MiniMiner" (2): mejora del movimiento, recoger objetos". http://www.nachocabanes.com/videojuegos/ipj/ipj36.php • Elaborar por equipos una solución de aplicación de videojuego de interés común para la generación de un producto de entretenimiento. • **Realizar la actividad de evaluación 2.2.1** Desarrolla aplicaciones de videojuegos donde se genere el código fuente, código ejecutable y pruebas de funcionamiento con la estructura siguiente: – Ciclo del videojuego –Entrada –Procesamiento –Salida – Finalización • Comentar en clase los resultados de la actividad de evaluación realizada, efectuando una coevaluación enfocada tanto al proceso ejecutado como a los resultados obtenidos.	

6. Prácticas/Ejercicios /Problemas/Actividades

Nombre del Alumno:		**Grupo:**	
Unidad de Aprendizaje 1:	Diseño de videojuegos		
Resultado de Aprendizaje:	**1.1** Determina los elementos según el desarrollo de videojuegos con base en tecnologías y herramientas de software.		
Actividad núm. 1:	Diferenciación de tipos de videojuegos, consolas y características.		

Elementos

1. Describe mediante una investigación lo siguiente:

- Entorno de desarrollo de videojuegos.
- Características de los videojuegos.
  - Funcionamiento en tiempo real.
  - Realización de tareas independientes al usuario - dibujar los objetos, actualizar coordenadas, calcular colisiones.
  - Trabajo bajo ciclos en espera de evento- respuesta.
- Estructura básica de un videojuego.
- Tipos de videojuegos utilizados en el mercado.
- Público hacia donde están dirigidos los videojuegos.

Herramientas

2. Describe las siguientes herramientas :

- Sistema operativo.
- Lenguajes de programación.
- Estructuras y bases de datos.
- Editores de texto.
- Archivos scripts.
- Comunicación de datos a través de redes.

- Bibliotecas.
- APIs Gráficas.
- Ventajas y desventajas de los lenguajes de desarrollo.

Diferenciación de Consolas para videojuego

3. Distingue mediante una investigación, lo siguiente::

- Ubicación por generación de cada consola y modelo.
- Características de cada consola y tecnologías adicionales que ocupa.
- Tipo de dispositivo de almacenamiento y de juego que soporta cada consola y la capacidad en bits.
- Ventajas o desventajas del uso de cada dispositivo.

4. Elabora una tabla comparativa de cada consola donde se plasmen las diferencias.

Nombre del Alumno:		**Grupo:**	
Unidad de Aprendizaje 1:	Diseño de videojuegos		
Resultado de Aprendizaje:	**1.2** Elabora el plan de proyecto del desarrollo de videojuegos mediante metodología específica, tareas de ejecución y ayuda de herramientas de software.		
Actividad núm. 2:	Propuesta del diseño de videojuego		

NOTA: El docente organizará equipos de trabajo, a fin de dar seguimiento posterior.

El desarrollo de aplicación se puede apoyar también en casos de videojuegos comerciales o de su interés.

Definición de objetivo
Define el objetivo de la aplicación de videojuego a desarrollar, de manera concreta, incluyendo los requerimientos específicos considerando:

- El asunto en torno al cual se realiza el proyecto y su alcance.
- Esquema o bosquejo de la idea en general.
- El objetivo definido de la propuesta de videojuego.
- Tipo de juego a hacer.

Determinación de tecnología y recursos a utilizar:

- Elementos de software necesarios.
- Disposición de equipo de cómputo.
- Utilización de dibujos, bosquejos, *story board* de las pantallas.
- Comportamiento o mecánica de los objetos del juego.
- Estudio de la sintaxis del lenguaje de programación a utilizar.
- El lenguaje de desarrollo a ocupar.
- Las herramientas de desarrollo para crear las imágenes.
- Las características técnicas del equipo de cómputo o dispositivo para el desarrollo.
- El equipo de personas de desarrollo o número de programadores.

Para elaborar un plan de trabajo debes reflexionar en torno a las siguientes preguntas.

- ¿Cuál es el objetivo de mi proyecto?
- ¿Qué actividades específicas necesito realizar para alcanzar los objetivos?
- ¿En qué secuencia se llevarán a cabo?
- ¿Cuándo estará terminada cada actividad?
- ¿Qué productos concretos obtendré en cada actividad?
- ¿Qué recursos necesito?

1. Escribe el tema de tu proyecto.
2. Escribe el objetivo (s).
3. Enlista las actividades.
4. Determina etapas y fechas de inicio y fin.
5. Determina los recursos humanos y materiales que se requieren y los productos a obtener.
6. Planifica a detalle una propuesta de proyecto de videojuego, incluyendo sus fases:
   - Concepción de la idea del juego.
   - Diseño.
   - Planificación.
   - Producción.
   - Pruebas.
7. Realiza el plan de actividades en un cronograma y revisa continuamente para retroalimentación. (Puedes utilizar el formato que se presenta a continuación):

ETAPA	ACTIVIDADES	FECHA INICIO	FECHA TERMINO	RECURSOS	PRODUCTOS
	1. 2.				
	1. 2. 3. 4.				

8. Grafica las fases del proyecto.

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Uso del modo gráfico y dibujando con herramientas del lenguaje.	Número:	1
Propósito de la práctica:	Elabora una aplicación de videojuego donde use el modo gráfico con herramientas del lenguaje, para dibujar elementos habituales (líneas, rectángulos, círculos, puntos, etc).		
Escenario:	Laboratorio de Informática	Duración	2 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	**Desempeños**
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows 2000 en adelante • Compilador de lenguaje seleccionado. (C, C++ o Java)	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj02.php **Pautas generales.** – Selecciona un lenguaje de programación. – Entrar a modo gráfico, e indicar número de puntos en pantalla y numero de colores. (cuantos más puntos y colores en pantalla, se necesitará una computadora más rápida y más habilidad de programación). – Debes adaptarte a las posibilidades del modo grafico de la computadora "cliente" de tu juego. – Una vez en modo gráfico, identifica órdenes preparadas y crear rutinas propias para casos

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	concretos. **Selección de órdenes habituales:** – Usa bibliotecas gráficas, para dibujar líneas, rectángulos o escribir textos, hasta opciones avanzadas para representación y manipulación de imágenes planas y de figuras en tres dimensiones. – Usa otras opciones del lenguaje escogido, desde las órdenes "habituales" para controlar el flujo de un programa (si..entonces, repetir..hasta) hasta órdenes más específicas (como las que generan números al azar). **Construcción de algoritmo de dibujo de elementos** – Utiliza el modo gráfico más estándar del lenguaje seleccionado. – Establece orden de Iniciación. – Usa la función de entrada a modo gráfico. – Dibuja con funciones línea entre dos puntos, con coordenadas verticales, horizontales y color. – Termina programa y sal del modo gráfico. **Presentación de código** – Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación. compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital. – Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores. Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.  **ADVERTENCIA DE RIESGO ELÉCTRICO**

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Leer del teclado y escribir texto usando lenguaje de programación	Número:	2
Propósito de la práctica:	Elabora una aplicación de videojuego que lea el teclado y escriba texto con herramientas del lenguaje, para emitir una frase simple a una compleja con características de tipo de letra, tamaño. Color y posición en la pantalla gráfica.		
Escenario:	Laboratorio de Informática	Duración	2 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows • Compilador de lenguaje seleccionado. (C, C++ o Java)	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj03.php **Pautas generales** – Selecciona un lenguaje de programación. – Selecciona una orden capaz de escribir un cierto texto en unas ciertas coordenadas. – Escribe una frase con tipos de letras (limitados al tipo de letra de la BIOS de la tarjeta gráfica o o del sistema operativo o que incorpora el compilador). – Usa funciones "prefabricadas". – Recuerda que para leer el teclado se debe tener una función que comprueba si se ha pulsado una tecla o no (para no tener que esperar) y otra que nos dice qué tecla se ha pulsado.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	– Define ciertas constantes para no tener que memorizar el número asociado a una cierta tecla, o el código ASCII que devuelve. **Construir un algoritmo para escribir texto** – Utiliza el lenguaje seleccionado y bibilotecas necesarias. – Utliza la función de biblioteca gráfica que permita escribir un mensaje en ciertas coordenadas. – Entra a modo gráfico. – Define el primer parámetro (usando "screen", la pantalla). – Establece la fuente de tipo de letra. – Indica el texto a escribir, las coordenadas x e y, y termina con el color y posición del texto. – Termina el programa y sal del modo gráfico. **Presentación de código** – Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación, compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital. – Compila el programa y genera su código ejecutable haciendo la corrida hasta que funcione sin errores. Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.  **ADVERTENCIA DE RIESGO ELÉCTRICO**

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Generar números al azar con método al aleatorio	Número:	3
Propósito de la práctica:	Elabora una aplicación de videojuego que genere números al azar con método al aleatorio para que a partir de esos números, con el juego se tenga que adivinar un número oculto, teniendo una cantidad limitada de intentos.		
Escenario:	Laboratorio de Informática	Duración	2 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows • Compilador de lenguaje seleccionado. (C, C++ o Java).	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj04.php **Pautas generales** – Considera cosas al azar con el fin de que el jugador tenga que enfrentarse a retos que no serán siempre los mismos. – Selecciona un lenguaje de programación. – Selecciona las funciones que dan ese número aleatorio ("rand"o "random"). – Si el lenguaje lo requiere da una "semilla" para empezar a generar los números. (semilla a partir del reloj).

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	Pseudocódigo del juego: Genera un número al azar entre 0 y 99 acertado = FALSO Pide un número al usuario Si el número tecleado es el número al azar terminado = VERDADERO Si el número tecleado es más pequeño, avisar Si el número tecleado es mayor, avisar Incrementa el número de intentos Hasta que (acertado) o (número de intentos = máximo) Si es acertado, felicitar En caso contrario decir qué número era el correcto **Construir un algoritmo para adivinar números** – Utiliza el lenguaje seleccionado y bibilotecas necesarias. – Entra a modo gráfico. – En base al pseudocódigo anterior construye un algoritmo generador de números al azar. – Sigue el pseudocódigo de antes pero adaptalo a la sintaxis del lenguaje. – Termina programa y sal del modo gráfico. **Presentación de código** – Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación.compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital. – Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores. Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.  **ADVERTENCIA DE RIESGO ELÉCTRICO**

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Elaborar el juego del ahorcado con método al aleatorio.	Número:	4
Propósito de la práctica:	Elabora una aplicación de videojuego genere un número al azar con método al aleatorio, para que con ese número se seleccione una palabra de entre las predefinidas y sea adivinada por el usuario.		
Escenario:	Laboratorio de Informática	Duración	2 horas

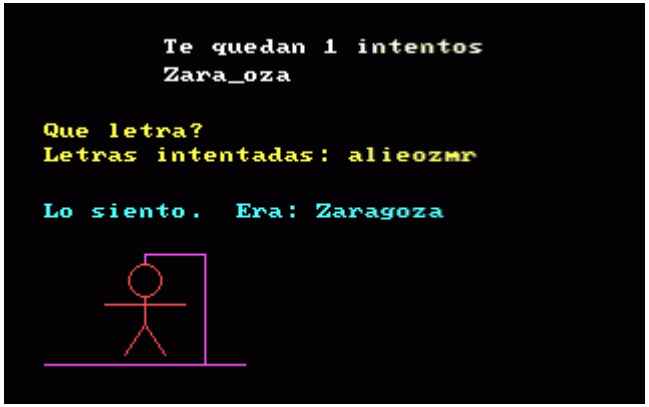
Materiales, Herramientas, Instrumental, Maquinaria y Equipo	**Desempeños**
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows • Compilador de lenguaje seleccionado. (C, C++ o Java).	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj05.php **Pautas generales.** – Pide al usuario elija una letra, si esa letra está en la palabra buscada, se muestra la posición donde se encuentra, sino agota los intentos . Hasta que se quede sin intentos o acierte la palabra. – Selecciona un lenguaje de programación. – Al tratarse de un juego gráfico, en cada "pasada" comprobar los fallos. – Dibuja con apariencia sencilla la parte del "patíbulo" del ahorcado.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños

Construir un algoritmo para el juego de ahorcado.

- Utiliza el lenguaje seleccionado y bibliotecas necesarias.
- Abre el modo gráfico.
  - Genera un número al azar donde:
    - El usuario elige una letra.
    - Si esa letra está en la palabra buscada, se muestra en que posición se encuentra.
    - Si no está en la palabra, le queda un intento menos.
  - Hasta que se quede sin intentos o acierte la palabra.
  - Si termina sin acertar, se indica cual era la palabra buscada.
  - En la parte gráfica por cada "pasada" comprueba cuantos fallos lleva.
  - Dibuja la correspondiente parte del "patíbulo" del ahorcado.

Algo así:



  - Termina el programa y sal del modo gráfico.

Presentación de código

- Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación, compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital.
- Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	sin errores. Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.  **ADVERTENCIA DE RIESGO ELÉCTRICO**

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Elaborar juego evitando esperar al teclado, motos de luz	Número:	5
Propósito de la práctica:	Elabora una aplicación de videojuego que evite esperar al teclado, para evitar que la computadora quede "parada " esperando a que se pulse una tecla y la acción prosiga aunque no se toque el teclado.		
Escenario:	Laboratorio de Informática	Duración	2 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	**Desempeños**
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows • Compilador de lenguaje seleccionado. (C, C++ o Java)	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj06.php **Pautas generales.** – Selecciona un lenguaje de programación. – Utiliza rutinas del compilador como la función **kbhit()**, que devuelve "verdadero" (distinto de cero) si se ha pulsado alguna tecla y "falso" (cero) si no se ha pulsado ninguna. – Utiliza la función **getch( ) o keypressed( )**, para saber exactamente qué tecla es la que se ha pulsado.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	– Vacía el "buffer" (memoria intermedia) del teclado para seguir comprobando nuevas teclas. **Construir un algoritmo para el juego "motos de luz".** – Utiliza el lenguaje seleccionado y bibliotecas necesarias. – Entra al modo gráfico. – Incluye como primera clausula función para tener acceso a rutinas de manejo de teclado "install_keyboard()". – Elabora la rutina secuencial de un juego con las siguientes condiciones: • Participan dos jugadores. • Cada uno de los jugadores maneja una "moto", que va dejando un rastro cuando se desplaza. • Si una de las motos "choca" con el rastro dejado por la otra, o por ella misma, o con el borde de la pantalla, estalla y pierde la partida. • Por tanto, el objetivo es "arrinconar" al contrario sin ser arrinconado antes por él. – Debes basarte en el siguiente pseudocódigo: • Borra la pantalla y dibujar un recuadro alrededor de ella. • Prepara las variables iniciales: dirección de cada jugador (incremento en X e Y de la posición de cada uno). • En el instante inicial, las motos se desplazarán en direcciones contrarias. En un juego "más real", este incremento podría ser variable (o disminuir el tiempo de pausa entre un movimiento y otro), de modo que las motos circularan cada vez más deprisa, para mayor dificultad. • Parte repetitiva: ○ Si la siguiente posición de alguna de las motos está ocupada, la moto estalla y el jugador pierde, termina el juego. ○ Desplazar las motos a su siguiente posición. ○ Si se pulsa alguna tecla, analizar esta tecla y cambiar la dirección de la moto según corresponda. ○ Pausa fija (o variable según la dificultad) antes de la siguiente repetición, fijada por el programa, para que la velocidad no dependa de la rapidez de la computadora que se utilice. • Repite indefinidamente (la condición de salida la comprobamos "dentro"). • Establece el choque de una moto otra, mirar el punto de la pantalla al que se va a mover. Si ese punto tiene un color distinto del fondo, hay choque (quizá sea con el borde de la pantalla, quizá con la estela, quizá con la del otro jugador) – Termina el programa y sal del modo gráfico.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	La apariencia del juego es, así:  **Presentación de código** – Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación.compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital. – Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores. Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias. **ADVERTENCIA DE RIESGO ELÉCTRICO**

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Crea juego Mini Serpiente 1 con uso de Mapas.	Número:	6
Propósito de la práctica:	Elabora una aplicación de videojuego con uso de Mapas para que una figura avance por la pantalla; si choca con la pared exterior, con su propia "cola" o con algún otro obstáculo, muere.		
Escenario:	Laboratorio de Informática	Duración	3 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	**Desempeños**
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows • Compilador de lenguaje seleccionado. (C, C++ o Java)	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj07.php **Pautas generales.** – Selecciona un lenguaje de programación. – Utiliza el mapa. – Dibuja obstáculos, el fondo y la serpiente con imágenes vistosas. – Toma de base el juego de las motos de luz, la idea básica coincide: el objeto se debe seguir moviendo aunque no se toque ninguna tecla, y en cada paso se debe comprobar si ha chocado con algún obstáculo.

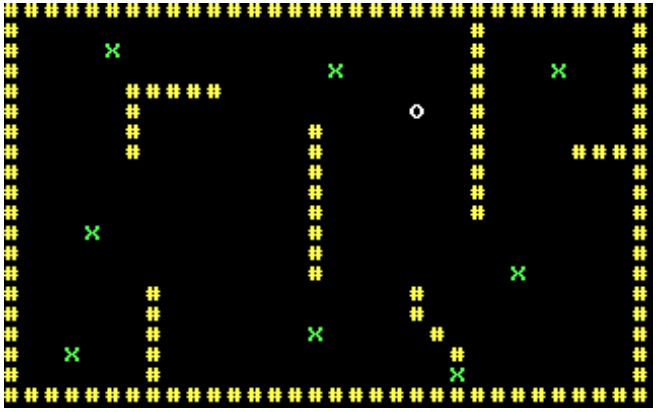
Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños

- A esta base añade el uso de un mapa.

Construir un algoritmo para miniserpiente 1

- Utiliza el lenguaje seleccionado y bibliotecas necesarias.
- Emplea el modo gráfico.
- Elabora la secuencia de algoritmos para crear una serpiente que avance por la pantalla; si choca con la pared exterior, con su propia "cola" o con algún otro obstáculo, muere.
- Memoriza un mapa con las posibles posiciones de la pantalla, e indicando cuales están vacías, cuales ocupadas por obstáculos y cuales ocupadas por comida.
- Utiliza funciones para el caso de uso de hilos.
- Termina el programa y sal del modo gráfico.

La apariencia del juego es la siguiente:



Presentación de código

- *Presenta las rutinas de programación* escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación, compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital.
- Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.  **ADVERTENCIA DE RIESGO ELÉCTRICO**

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Crear juego miniserpiente 2 usando lenguaje de programación	Número:	7
Propósito de la práctica:	Elabora una aplicación de videojuego con figuras multicolor usando lenguaje de programación para que las figuras se muevan		
Escenario:	Laboratorio de Informática	Duración	2 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows • Compilador de lenguaje seleccionado. (C, C++ o Java)	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj08.php **Pautas generales.** – Selecciona un lenguaje de programación. – Utiliza "sprites" para crear figuras "transparentes": con "huecos" alrededor o en su interior y a través de estos huecos donde se ve el fondo. **Construir un algoritmo para juego de Miniserpiente 2** – Utiliza el lenguaje seleccionado y bibliotecas necesarias. – Emplea el modo gráfico.

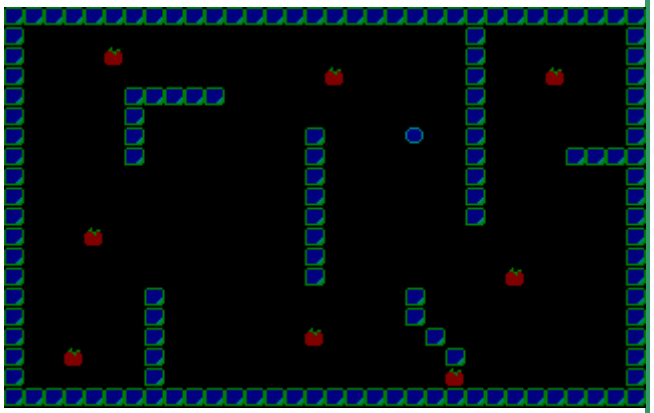
Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños

- Para el manejo de "**sprites**"., el pseudocódigo en Pascal de esta función es:

```
procedimiento dibujar_sprite(imagen, posicX, posicY);
      para i = 1 hasta ANCHOSPRITE
      para j = 1 hasta ALTOSPRITE
      si imagen[j,i] <> 0 entonces
      dibujar_punto(x+i-1, y+j-1, imagen[j,i]);
```

- Elabora rutinas algorítmicas donde se construya el juego con figuras multicolor que se muevan
- Declara arreglos o “array” de dos dimensiones, que indicará el color de cada uno de los puntos que forman la figura, que indicará el color de cada uno de los puntos que forman la figura:

```
spriteLadrillo =

{{0,2,2,2,2,2,2,2,2,0},

 {2,1,1,1,1,1,1,1,1,2},

 {2,1,1,1,1,1,1,1,1,2},

 {2,1,1,1,1,1,1,1,1,2},

 {2,1,1,1,1,1,1,1,1,2},

 {2,1,1,1,1,1,1,1,3,2},

 {2,1,1,1,1,1,1,1,3,3,2},

 {2,2,2,2,2,2,2,2,2,0}

 };
```



- Hacer usos de funciones y órdenes necesarias propias del lenguaje.
- Termina el programa y salir de modo gráfico.

Presentación de código

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	– Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación, compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital. – Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores. Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias. **ADVERTENCIA DE RIESGO ELÉCTRICO**

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Evitar los parpadeos, juego minisejrpiente 3 mediante aproximaciones	Número:	8
Propósito de la práctica:	Elabora una aplicación de videojuego mediante aproximaciones, para que se eviten parpadeos al avnzar.		
Escenario:	Laboratorio de Informática	Duración	2 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	**Desempeños**
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows Compilador de lenguaje seleccionado. (C, C++ o Java)	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj09.php **Pautas generales.** – Selecciona un lenguaje de programación. – Utiliza aproximaciones con técnica "**doble buffer**" que consiste en no escribir directamente en pantalla, sino dibujar lo que va a aparecer, en una "pantalla no visible". **Construir un algoritmo para juego de Miniserpiente 3** – Utiliza el lenguaje seleccionado y bibilotecas necesarias.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	– Emplea el modo gráfico. – Crea las rutinas algorítmicas necesarias para un juego donde se eviten parpadeos mediante aproximaciones. – Declara rutinas para hacer aproximaciones mediante empleo de "**doble buffer**". – Crea un nuevo bitmap llamado "pantallaOculta", se dibuja ahí la nave, los marcianos y el disparo. – Vuelca en pantalla al final de cada secuencia completa de dibujado. – Vuelve a dibujar aunque sea realmente lo mismo que ya había antes – Termina el programa y sal del modo gráfico. **Presentación de código** – Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación, compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital. – Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores. Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias. **ADVERTENCIA DE RIESGO ELÉCTRICO**

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Crear juego usando la paleta de colores	Número:	9
Propósito de la práctica:	Elabora una aplicación de videojuego que use paletas de colores donde se manejen 256 colores "estándar" para trabajar con una gama de colores posibles de la computadora.		
Escenario:	Laboratorio de Informática	Duración	2 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows. • Compilador de lenguaje seleccionado. (C, C++ o Java),	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj10.php **Pautas generales.** – Selecciona un lenguaje de programación. – Utiliza bibliotecas gráficas. – Abarca los 16 colores estándar en cualquier computadora "tipo PC" que tenga pantalla a color (tarjeta gráfica CGA o superiores). Estos colores son los que se emplean en una pantalla de texto en el modo normal de 16 colores:

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños

Números de color	Equivale a
0	Negro
1	Azul
2	Verde
3	Cyan
4	Rojo
5	Violeta
6	Marrón
7	Gris claro
8	Gris oscuro
9	Azul claro
10	Verde claro
11	Cyan claro
12	Rojo claro
13	Violeta claro
14	Amarillo
15	Blanco

Construir un algoritmo para empleo de paleta de colores

- Utiliza el lenguaje seleccionado y bibilotecas necesarias
- Emplea el modo gráfico.
- Elabora rutinas algorítmicas donde se utilizen millones de colores simultáneamente en la pantalla, capaz de mostrar 256 colore simultáneos, escogidos de una paleta de más de 250.000.

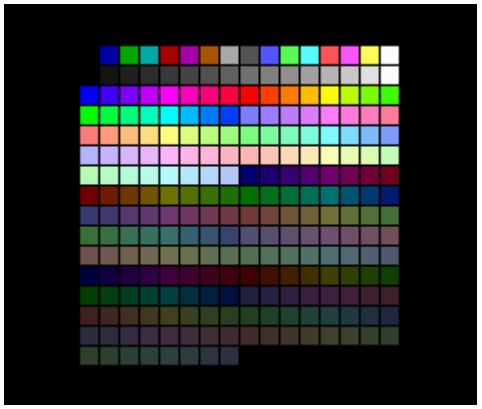
La mayoría de los compiladores de lenguajes como C y Pascal tienen definidas unas constantes (en inglés):

Black = 0;
Blue = 1;
Green = 2;

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños

Cyan = 3;
Red = 4;
Magenta = 5;
Brown = 6;
LightGray = 7;
DarkGray = 8;
LightBlue = 9;
LightGreen = 10;
LightCyan = 11;
LightRed = 12;
LightMagenta = 13;
Yellow = 14;
White = 15;

- Crea un pequeño programa que muestre la apariencia de estos 16 colores y los otros 240 que forman la " **paleta estándar VGA** ", podemos. Por ejemplo, en 16 filas y 16 columnas, dibujando "recuadritos" de 8 puntos de ancho y 8 puntos de alto, así:



- Modifica el color con bibliotecas gráficas, indicando qué cantidad de rojo, verde y azul se requiere que lo compongan. Se tienen 64 tonalidades posibles de cada uno de los colores básicos, lo que permite utilizar 262.144 tonos de color distintos.
- Se puede jugar con toda la paleta de colores: memoriza la paleta de colores actual (para poder modificarla y recuperarla después), oscurece la paleta, convirtiéndola gradualmente en otra.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	– Termina el programa y sal del modo gráfico. **Presentación de código** – Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación.compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital. – Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores. Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias. **ADVERTENCIA DE RIESGO ELÉCTRICO**

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Crear juego usando funciones del mouse	Número:	10
Propósito de la práctica:	Elabora una aplicación de videojuego que use funciones del mouse para comprobar si se ha hecho un doble clic, o cambiar la forma del puntero, o leer la posición de la rueda o limitar el movimiento a ciertas zonas de la pantalla.		
Escenario:	Laboratorio de Informática	Duración	2 horas

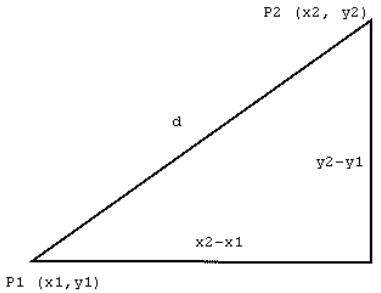
Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows • Compilador de lenguaje seleccionado. (C, C++ o Java)	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj12.php **Pautas generales.** – Selecciona un lenguaje de programación. – Utiliza bibliotecas de funciones ya creadas para el manejo de mouse. **Construir un algoritmo para el juego Puntería** – Utiliza el lenguaje seleccionado y bibilotecas necesarias. – Emplea el modo gráfico. – Realiza el llamado a bibliotecas de uso del mouse:

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	○ Mostrar el ratón ("showMouse"). ○ Ocultar el ratón (hideMouse). ○ Leer en qué posición se encuentra (getMouseX y getMouseY), ○ Mover el ratón a cierto punto (setMouseX y setMouseY). ○ vVr si se ha pulsado un cierto botón del ratón (getMouseButton). – Otras posibilidades más avanzadas, como comprobar si se ha hecho un doble clic, o cambiar la forma del puntero, o leer la posición de la rueda (si nuestro ratón tiene una y el driver lo reconoce), o limitar el movimiento a ciertas zonas de la pantalla, pueden no estar disponibles en algunas bibliotecas, y por supuesto los nombres exactos cambiarán de una biblioteca a otra, dependiendo del lenguaje. – Crea las rutinas algorítmicas necesarias para generar un juego donde se haga uso del mouse y la idea básica es: ○ Dibujar un recuadro en pantalla, con posición y tamaño al azar. ○ Comprobar si se pulsa el ratón. ○ Si se ha pulsado dentro del recuadro -> un punto más para el jugador. ○ En cualquier caso, dibujar otro rectángulo distinto. ○ Todo ello, repetido hasta que se pulse una tecla. – Considerar lo siguiente: ○ Que el juego acabe después de un cierto número de intentos, o de un cierto tiempo ○ Los puntos obtenidos dependen del tiempo que se tarde en acertar. ○ Hacer una diana, considerar el tamaño. ○ Comprobar que se pulse el mouse dentro de las zona de la diana. – Termina el programa y sal del modo gráfico. **Presentación de código** – Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación, compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital. – Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores. Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.  **ADVERTENCIA DE RIESGO ELÉCTRICO**

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Aplicación de matemáticas para Juego: TiroAlPlato. usando lenguaje de programación	Número:	11
Propósito de la práctica:	Elabora una aplicación de videojuego de TiroAlplato usando lenguaje de programación para la aplicación de conceptos de geometría plana.		
Escenario:	Laboratorio de Informática	Duración	2 horas

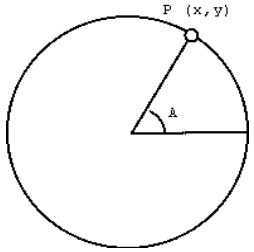
Materiales, Herramientas, Instrumental, Maquinaria y Equipo	**Desempeños**
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows • Compilador de lenguaje seleccionado. (C, C++ o Java)	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj13.php **Pautas generales.** – Selecciona un lenguaje de programación. – Debes contar con conocimientos matemáticos de geometría (distancias, círculos, parábolas y su utilización). **Ecuación de Distancias**. Teorema de Pitágoras, donde "el cuadrado de la hipotenusa es igual a la suma de los cuadrados de los catetos". Si unimos dos puntos con una línea recta, podríamos formar un triángulo rectángulo: su hipotenusa sería la recta que une los dos puntos,

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	el tamaño de un cateto sería la diferencia entre las coordenadas horizontales (x) de los puntos, y el tamaño del otro cateto sería la diferencia entre las coordenadas verticales (y):



la forma de hallar la distancia entre dos puntos es: $d = raíz\ (\ (x_2 - x_1)^2 + (y_2 - y_1)^2\ )$.

Círculo. La ecuación que nos da las coordenadas de los puntos de una circunferencia a partir del radio de la circunferencia y del ángulo en que se encuentra ese punto son:

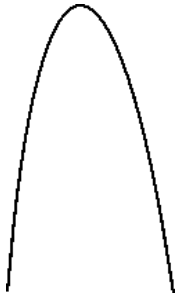


x = radio * coseno (ángulo)
y = radio * seno (ángulo)

Nota: eso es en el caso de que el centro sea el punto (0,0); si no lo fuera, basta con sumar a estos valores las coordenadas x e y del centro del círculo -

Parábola La forma general de un parábola es $y = ax^2 + bx + c$. Los valores de a, b, y c dependen de cómo esté de desplazada la parábola y de lo "afilada" que sea.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños



- Una parábola como la de la imagen anterior debe tener el coeficiente "a" (el número que acompaña a x^2) negativo; si es un número positivo, la parábola sería "al revés", con el hueco hacia arriba. Eso sí, en la pantalla de la computadora, las coordenadas verticales (y) se suelen medir de arriba a abajo, de modo que esa ecuación será la útil para nosotros, tal y como aparece.

Para conseguir que la parábola pase por un cierto punto, se necesita que tenga su vértice en el punto (x1, y1) y que tenga una distancia "p" desde dicho vértice hasta un punto especial llamado "foco" (que está "dentro" de la parábola, en la misma vertical que el vértice. $(x-x_1)^2 = 2p\ (y-y_1)$.

Podemos desarrollar esta expresión y obtener cuanto tendría que ser la "a", la "b" y la "c" a partir de las coordenadas del vértice (x1, y1) y de la distancia "p" (cuanto menor sea p, más "abierta" será la parábola):

$$a = 1 / 2p$$
$$b = -x_1 / p$$
$$c = (x_1{}^2 / 2p) + y_1$$

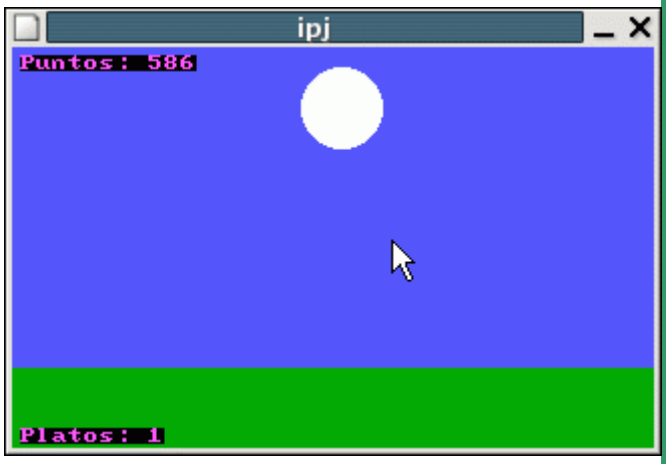
Construir un algoritmo para Juego TiroAlPlato

- Utiliza el lenguaje seleccionado y bibilotecas necesarias.
- Emplea el modo gráfico.
- Elabora las rutinas necesarias para el juego de uso del mouse donde los blancos estarán en movimiento, siguiendo una curva que será una parábola, y serán circulares. La puntuación dependerá del tiempo que se tarde en acertar. Habrá un número limitado de "platos", tras el cual se acabará la partida.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños

Con todo esto, la mecánica del juego será:

```
 Inicializar variables
 Repetir para cada plato:
    Dibujar plato a la izqda de la
pantalla
    Repetir
      Si se pulsa el ratón en plato
        Aumentar puntuación
      Si no, al cabo de un tiempo
        Calcular nueva posición del
plato
        Redibujar
    Hasta que el plato salga (dcha) o
se acierte
 Hasta que se acaben los platos
 Mostrar puntuación final
```



- Se puede usar un blanco redondo en vez de rectangular.
- Se comprueba si se ha acertado viendo si la distancia desde el centro del círculo hasta el punto en el que se ha pulsado el mouse es menor (o igual) que el radio del círculo.
- Usa expresiones para dibujar un círculo, para rotar un objeto en el plano.
- Termina el programa y sal del modo gráfico.

Presentación de código

- Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital.
- Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores.

Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.
Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.

ADVERTENCIA DE RIESGO ELÉCTRICO

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Reproducir sonidos. Juego SimeonDice usando lenguaje de programación	Número:	12
Propósito de la práctica:	Elabora una aplicación de videojuego SimeonDice usando lenguaje de programación para generar el juego clásico de memoria auditiva donde se hacen reproducciones de sonido.		
Escenario:	Laboratorio de Informática	Duración	2 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows • Compilador de lenguaje seleccionado. (C, C++ o Java)	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj14.php **Pautas generales.** – Selecciona un lenguaje de programación. – Se puede hacer uso de altavoz de la PC. – Utiliza bibliotecas de funciones para sonido Recuerda que los principales tipos de sonidos por computadora que sirven para juegos y

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños

reproducirlos son los siguientes:

1) Sonidos simples mediante el altavoz	Emitir un sonido de una cierta frecuencia durante un cierto tiempo. Se pueden usar las ordenes de emitir un sonido con la orden "sound(freq)" donde "freq" es la frecuencia del sonido . Paramos este sonido con "nosound". Para que el sonido dure un cierto tiempo, podemos usar "delay(ms)" para esperar una cantidad de milisegundos, o bien hacer otra cosa mientras se escucha el sonido y comprobar la hora continuamente. Existen estas órdenes para muchos compiladores de C como Turbo C y DJGPP (para DJGPP están declaradas en "pc.h"): sound(freq); … nosound().
2) Sonidos digitalizados.	Capturar un sonido con la ayuda de un micrófono y reproducirlo posteriormente. El estándar en Windows es el formato WAV, Otro formato conocido es el VOC, de Creative Labs, la casa desarrolladora de las tarjetas de sonido SoundBlaster. Y un tercer formato, es el AU. Hay herramientas para convertir de un formato a otro. Un cuarto formato es el formato MP3, que es un formato comprimido, lo que supone una cierta pérdida de calidad a cambio de ocupar menos espacio (habitualmente cerca de un 10% de lo que ocuparía el WAV correspondiente). Usar secuencia de órdenes para capturar un sonido con la Grabadora de Windows, guardarlo en formato WAV y reproducirlo. En la orden "play_sample" los parámetros que se indican son: el sonido a reproducir, el volumen (0 a 255), el desplazamiento (posición a partir de la que reproducir, "pan", 0 a 255), la frecuencia (relativa: 1000 es la velocidad original, 2000 es el doble y así sucesivamente) y si se debe repetir (TRUE para si y FALSE para no repetir). Si un sonido se está repitiendo, para pararlo se usa "stop_sample".
3) Melodías MIDI.	Son melodías formadas por secuencias de notas. Se crean con programas que muestran partituras en pantalla, o incluso conectando ciertos instrumentos musicales electrónicos (teclados) a una conexión que muchos computadoras incluyen (MIDI: Musical Instrument Device Interface). Entornos como Windows (e incluso muchos teléfonos móviles actuales) incluyen reproductores de sonido capaces de manejar ficheros **MID** . Se puede reproducir los sonidos con la orden "play_midi", "play_sample".
4) Melodías MOD y similares (S3M, XM, etc).	Son formatos más complejos que el anterior, ya que junto con las notas a reproducir se detalla cómo "suena" cada uno de los instrumentos (normalmente usando una "muestra" digitalizada -en inglés: "sample").

Construir un algoritmo para Juego Simón dice

- Utiliza el lenguaje seleccionado y bibilotecas necesarias.
- Emplea el modo gráfico.

  El juego consiste en repetir la secuencia de sonidos (y luces) que nos va proponiendo la computadora: primero será una sola nota; si la recordamos, se añade un segunda nota; si recordamos estas dos se añade una tercera, después una cuarta y así sucesivamente hasta que cometamos un error.
- Genera los algoritmos necesarios para la elaboración del juego basándose en el siguiente

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños

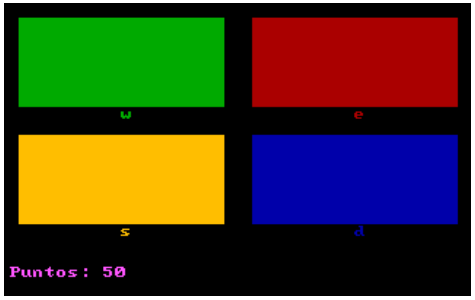
pseudocódigo:

```
secuenciaDeNotas = vacia
repetir
  generar nuevaNota al azar
  anadir nuevaNota a secuenciaDeNotas
  reproducir secuenciaDeNotas
  fallo = FALSO
  desde i = 1 hasta (i=numeroDeNotas) o hasta (fallo)
    esperar a que el usuario escoja una nota
    si notaEscogida != nota[i] entonces fallo=VERDADERO
  finDesde
hasta fallo
puntuacion = numeroNotasAcertadas * 10
```

- El tablero puede ser que debajo de cada color se recuerde la tecla que se debe pulsar:

  Nota: Para reproducir notas con formato MIDI se necesita algún editor de melodías MIDI

- Terminar programa y salir de modo gráfico.



Presentación de código

- Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación.compuesto por instrucciones en líneas de texto de los programas o algoritmos

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	elaborados, de manera impresa y/o digital. – Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores. Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.  **ADVERTENCIA DE RIESGO ELÉCTRICO**

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Uso de formatos de archivos de imágenes usando lenguaje de programación.	Número:	13
Propósito de la práctica:	Elabora una aplicación de videojuego donde use formato de archivos para hacer manipulación de imágenes mediante lectura de imágenes desde archivos.		
Escenario:	Laboratorio de Informática	Duración	2 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	**Desempeños**
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows • Compilador de lenguaje seleccionado. (C, C++ o Java)	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj15.php **Pautas generales.** – Selecciona un lenguaje de programación – Recuerda que las imágenes creadas por computadora, se distinguen en dos grandes grupos: • **Bitmap**: (o mapa de bits). Un tipo de imágenes para computadora, en las que se almacena información sobre los puntos que las componen y el color de cada punto. Esto supone que al ampliar la imagen no se gana en definición, sino que se ven "puntos gordos". • **Vectorial**: Un tipo de imágenes para computadora, en las que se almacena información

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	sobre las líneas y figuras geométricas que las componen. Esto permite que no pierdan definición si se amplían, al contrario de lo que ocurre con las imágenes "Bitmap". – Utiliza imágenes Bitmap, que se diseñan con un programa de manipulación de imágenes, se guardan en un cierto formato y desde el programa se cargan y manipulan. Los formatos de imagen Bitmap son: • **JPG** o JPEG: Es un formato comprimido, pero que PIERDE CALIDAD en la compresión. Se puede indicar el grado de compresión (o de calidad) que se desea, para conseguir un tamaño más pequeño o bien una mayor fidelidad al original. Por ejemplo, una compresión del 15% (o una calidad del 85%) es un valor que permite imágenes con bastante calidad pero con un tamaño no muy grande. • **GIF**: Es un formato comprimido, que no puede mostrar más de 256 colores, pero que no pierde calidad en la compresión. También permite otras características "avanzadas", como incluir varias imágenes en un mismo fichero (muy utilizado para crear animaciones sencillas en Internet) o definir un color transparente (que no se dibujará en pantalla al mostrar la imagen, también muy usado en Internet). Como inconveniente adicional, es un formato que tiene copyright, por lo que algunas utilidades no permiten usarlo. • **TIFF**: Es un formato con que no pierde calidad y que se usa con frecuencia para intercambiar información entre computadoras muy distintos (PC y Macintosh, por ejemplo) cuando se debe conservar toda la calidad de la imagen original. – Existen distintas variantes, unas sin comprimir y otras comprimidas, pero aun así comprime la información mucho menos que el formato JPG • **BMP**: Formato nativo de Windows. Permite desde imágenes en blanco y negro hasta "color auténtico", y existe también una variante sin comprimir y una variante comprimida (que, al igual que los TIFF, ocupa mucho más que el JPG). • **PCX**: Formato diseñado por los creadores de **Paintbrush** (un programa de dibujo que se utilizó mucho en los tiempos de MsDos). Es comprimido, pero compacta la información menos que el formato GIF. Hoy en día este formato se usa poco, pero es un formato muy sencillo, por lo que es habitual encontrar bibliotecas de funciones para juegos que lo utilizan. • **LBM**: Es un caso muy similar. Se trata del formato usado por el antiguo programa de dibujo **Deluxe Paint.** • **PNG** : Diseñado como alternativa a GIF cuando los creadores de este formato decidieron cobrar royalties por su uso. Además incluye alguna mejora, como el soporte para imágenes de más de 256 colores.Se utiliza en Internet, pero no está tan extendido como GIF. • **TGA**: Es un formato sencillo y que permite imágenes de una alta calidad (separa los componentes rojo, verde y azul de cada punto de la imagen) pero a cambio de ocupar mucho espacio en disco.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	• **PSD**: Formato nativo de Photoshop, uno de los programa de retoque de imágenes más utilizados (los ficheros no podrán ser visualizados desde muchas otras aplicaciones). • **CPT**: Formato nativo de Corel Photopaint, otro conocido programa de retoque (con el mismo inconveniente que el anterior). **Construir un algoritmo para uso de formatos de archivos de imágnes** – Utiliza el lenguaje seleccionado y bibilotecas necesarias. – Emplea el modo gráfico. – Elabora las rutinas de algoritmos necesarios para hacer manejar y hacer llamadas a imágenes prediseñadas en archivos específicos. – Realiza llamadas a archivos de imagen con las consideraciones de: • Usar archivos que la plataforma (sistema operativo) los soporte directamente, imágenes BMP. • Usar otros formatos frecuentes, como JPG o GIF que se cargan mediante bibliotecas. Formatos como el PCX creando rutinas para mostrarlos. – Terminar programa y salir de modo gráfico. **Presentación de código** – Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación.compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital. – Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores. Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.  **ADVERTENCIA DE RIESGO ELÉCTRICO**

Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Movimientos de acercamiento usando lenguaje de programación.	Número:	14
Propósito de la práctica:	Elabora una aplicación de videojuego “Space Invaders” usando lenguaje de programación para que muestre los movimientos mediante acercamientos.		
Escenario:	Laboratorio de Informática	Duración	5 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	**Desempeños**
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows • Compilador de lenguaje seleccionado. (C, C++ o Java)	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en la referencia siguiente: http://www.nachocabanes.com/videojuegos/ipj/ipj17.php http://www.nachocabanes.com/videojuegos/ipj/ipj18.php **Pautas generales.** – Selecciona un lenguaje de programación. **Construir un algoritmo para juego Marciano 1. (acercamiento)** – Utiliza el lenguaje seleccionado y bibliotecas necesarias. La idea del juego es la siguiente:

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños

- Manejar una nave, capaz de moverse a izquierda y derecha (por supuesto, no debe salir de la pantalla).
- En la parte superior de la pantalla habrá un "marciano", capaz de moverse de izquierda a derecha y de ir bajando paulatinamente hacia cierta posición. Sólo hay un enemigo y no es capaz de disparar.
- La nave sí debe ser capaz de disparar, y el programa ha de detectar cuando el disparo alcanza al enemigo, momento en el que termina la partida. Puede haber dos disparos en pantalla a la vez.
- El enemigo debe "cambiar de forma" a medida que se mueve, no ser totalmente "estático" (basta con dos "fotogramas").
- Los tres elementos (nave, enemigo y disparo deben ser capaces de moverse independientemente uno del otro, y a velocidades distintas).

La pantalla del juego puede ser así: (apariencia basada en la del clásico juego "Space Invaders"):

– La secuencia de pasos se establece:

1. Paso 1: Todavía no hay movimiento del "marciano" ni de la nave, limitarse ver cómo cargar las imágenes que se manejan en los otros 3 pasos.
2. Paso 2: Movimiento sólo del "marciano", de lado a lado, bajando en la pantalla y cambiando de forma. Se podrá interrumpir pulsando Esc. No debe salirse por los lados, y si llega a la parte inferior debe volver a la superior.
3. Paso 3: Añadir la nave, capaz de moverse de lado a lado, a velocidad distinta. Tampoco

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños

deber salir por los lados.

4. Paso 4: Finalmente, incluir la posibilidad de disparar, el movimiento vertical del disparo y la detección de colisiones entre el disparo y el enemigo. El disparo "desaparecerá" cuando golpee al enemigo o llegue a la parte superior de la pantalla, y sólo entonces se podrá volver a disparar.

– Cargar y mostrar imágenes leyendo desde un archivo.

–

- Las cifras del 0 al 9, para mostrar la puntuación.
- Los "letreros" que interesan en pantalla (SCORE=Puntuación, UP para formar 1UP= Primer jugador y 2UP=Segundo jugador, HI- para HI-SCORE= mejor puntuación).
- Los primeros "personajes": las dos apariencias posibles del "marciano", la nave, la "torre" del disparo.

– Elabora el algoritmo correspondiente, de acuerdo a la secuencias de pasos anterior.
– Las imágenes a mostrar deben tener un tamaño fijo.
– Termina el programa.

Construir un algoritmo para juego Marciano 2

– Utiliza el lenguaje seleccionado y bibliotecas necesarias.

La idea de este segundo juego de aproximación es: movimiento sólo del "marciano", de lado a lado, bajando en la pantalla y cambiando de forma. Se podrá interrumpir pulsando Esc. No deberá salirse por los lados, y si llega a la parte inferior deberá volver a la superior.

El marciano debe "cambiar de forma". Según el caso, el "personaje" puede cambiar de apariencia cada vez que lo dibujamos, de forma sucesiva, o mantener una misma forma durante varios fotogramas. En el siguiente pseudocódigo se toma la última opción, pero en su variante más sencilla: cada X fotogramas (y siempre los mismos, por ejemplo, cada 3) cambiaremos la apariencia del personaje y volveremos a empezar a contar:

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños

	cargar imagen 1 y 2 del marciano
	entrar a modo gráfico
	dibujar marciano en punto inicial
	repetir
	  aumentar posición horizontal
	  si se acerca a margen lateral de la pantalla
	  cambiar sentido horizontal de avance
	  bajar una línea en vertical
	  si se acerca a parte inferior de la pantalla
	  mover a parte superior
	  aumentar contador de "fotogramas"
	  si contador de fotogramas = numFotogramasPorImagen
	  cambiar apariencia de marciano
	  contador de fotogramas = 0
	  redibujar pantalla
	  pausa
	hasta que se pulse ESC
	
	La aproximación anterior es capaz de mostrar las imágenes, esta añade pocas novedades: calcular la nueva posición del marciano (cambiando de línea y de dirección o volviendo a la parte superior de la pantalla si corresponde) y salir sólo si se pulsa la tecla ESC, no con las demás teclas.
	– Verifica que se mueven dos cosas a velocidad distinta (el "marciano" y la nave).
	– Elabora los algoritmos correspondientes para el juego con las condiciones explicadas anteriormente en lenguaje de programación seleccionado.
	– Termina el programa.
	
	Presentación de código
	
	– Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital.
	– Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores.
	
	Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.  **ADVERTENCIA DE RIESGO ELÉCTRICO**

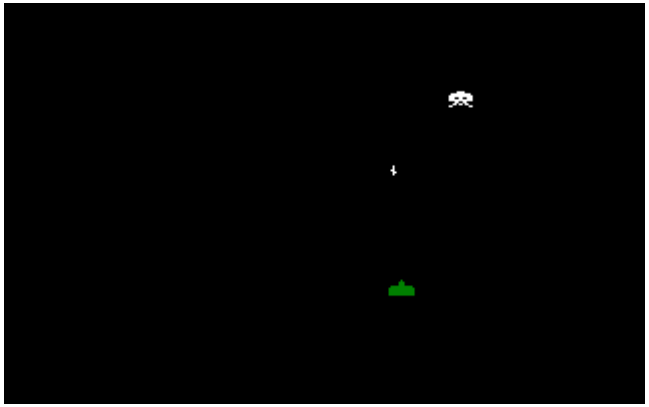
Unidad de Aprendizaje:	Diseño de videojuegos	Número:	2
Práctica:	Manejo de distintas aproximaciones usando lenguaje de programación	Número:	15
Propósito de la práctica:	Elabora una aplicación de videojuego “juego de Marciano” usando lenguaje de programación que muestre los movimientos mediante varios tipos de acercamientos.		
Escenario:	Laboratorio de Informática	Duración	10 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
• Equipo de cómputo Core Duo o superior • Dispositivo de almacenamiento (USB) • Windows • Compilador de lenguaje seleccionado. (C, C++ o Java)	• Aplica las siguientes medidas de seguridad e higiene en el desarrollo de la práctica: – Evita la manipulación de comida o líquidos cerca del equipo de cómputo – No introduce objetos extraños en las entradas físicas de dispositivos de la computadora – No utiliza imanes cerca de discos compactos, memorias extraíbles ó de la computadora – Limpia el área de trabajo, prepara herramientas y los materiales a utilizar ↺ Utilizar las hojas por ambas caras y colocar las de desecho las en el recipiente destinado para su posterior envió a reciclaje **NOTA al Alumno:** Realizar un respaldo de la información que generes en un dispositivo de almacenamiento. **NOTA** El docente deberá adecuar la práctica al equipo y recursos de software con el que se cuenta. **NOTA**: El docente organizará equipos de trabajo y puede basarse en las referencias siguientes: http://www.nachocabanes.com/videojuegos/ipj/ipj19.php http://www.nachocabanes.com/videojuegos/ipj/ipj20.php http://www.nachocabanes.com/videojuegos/ipj/ipj21.php http://www.nachocabanes.com/videojuegos/ipj/ipj22.php **Pautas generales.** – Selecciona un lenguaje de programación. **Construir un algoritmo para juego de Marciano 3**

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	– Utiliza el lenguaje seleccionado y bibliotecas necesarias. La idea de esta tercera aproximación es: añadir a la nave, capacidad de moverse de lado a lado, a velocidad distinta. Tampoco deberá poderse salir por los lados. La dificultad está en que se mueva a la vez que el marciano y a velocidad distinta de la de éste. Ahora no se puede usar "pausa" porque el programa no debe esperar, lo que se debe comprobar continuamente si ha llegado el momento de mover alguno de los dos personajes. El pseudocódigo es el siguiente: repetir si hayQueRedibujarMarciano dibujar marciano hayQueRedibujarMarciano = FALSO si hayQueRedibujarNave dibujar nave hayQueRedibujarNave = FALSO si tiempoTranscurridoNave = tiempoHastaMoverNave comprobar si se ha pulsado alguna tecla calcular nueva posición de nave hayQueRedibujarNave = VERDADERO si tiempoTranscurridoMarciano = tiempoHastaMoverMarciano calcular nueva posición de marciano hayQueRedibujarMarciano = VERDADERO hasta que se pulse ESC – Calcula nueva posición de marciano equivalente a varias órdenes (de hecho, a casi toda la aproximación anterior) y que expresiones como "tiempoTranscurridoMarciano" serían simplemente una resta entre dos tiempos, algo así como "HoraActual - HoraUltimoMovimientoMarciano" si tenemos un reloj disponible, o algo como "contador > RetardoEsperado" si tenemos que usar un contador.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	 – En base al pseudocódigo anterior elabora su algoritmo correspondiente en el lenguaje seleccionado. – Termina el programa. **Construir un algoritmo para juego de Marciano 4 Incluyendo un disparo y comprobación de colisiones.** – Utilizarel lenguaje seleccionado y bibilotecas necesarias. – Añade el disparo y la detección de colisiones. La idea de esta cuarta aproximación es: Incluir la posibilidad de disparar, el movimiento vertical del disparo y la detección de colisiones entre el disparo y el enemigo. El disparo "desaparecerá" cuando golpee al enemigo o llegue a la parte superior de la pantalla, y sólo entonces se podrá volver a disparar. – Añade un tercer "personaje" a controlar, el disparo, que no añade casi ninguna complicación, salvo por el hecho de que no siempre está "activo" (puede haber momentos en los que no estemos disparando). La novedad es cómo comprobar si el disparo ha impactado en la nave enemiga. Una forma muy sencilla de conseguirlo puede ser simplemente comparar la posición del disparo y la del marciano. Como también sabemos el "tamaño" del marciano (su anchura y su altura), podemos saber si el disparo ha acertado (si su coordenada X está entre XMarciano y XMarciano+AnchoMarciano, y lo mismo para la coordenada Y).

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños



Esta aproximación es similar a la aproximación anterior, aunque ligeramente más largo:

- En base a las condiciones anteriores elabora su algoritmo correspondiente en el lenguaje seleccionado.
- Termina el programa.

Construir un algoritmo para juego de Marciano 5 (Moviendo varios marcianos a la vez)

- Utiliza el lenguaje seleccionado y bibilotecas necesarias.

Ahora no hay un único enemigo, sino varios. Además, no siempre habrá que redibujar todos los enemigos, sino que algunos todavía estarán "vivos", mientras que a otros ya los habremos "matado". El juego terminará cuando matemos a todos los enemigos (en un juego "real" se debería pasar a un nivel de dificultad mayor) o cuando se nos mate a nosotros (en la práctica debería ser cuando agotáramos un cierto número de "vidas", habitualmente 3).

- Realiza nuevamente varias aproximaciones sucesivas hasta tener un juego "razonablemente jugable".

En este primer acercamiento, simplemente mostrará varios enemigos moviéndose a la vez.
Las diferencias entre tener un único enemigo o varios hay que considerar:

- Hay que mover varios enemigos a la vez. Esto se puede hacer guardando la información de todos ellos dentro de un "array", tanto en nuestro caso, en el que todos los enemigos se moverán a la vez, como en el caso de que tuvieran movimientos relativamente independientes. Para cada marciano puede interesar (según el caso) guardar información

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	como: posición actual, trayectoria actual, apariencia actual, etcétera. • Se puede ir "matando marcianos", de modo que no estarán siempre todos visibles. Se puede eliminar realmente a los marcianos (pasaríamos de tener "n" a tener "n-1") o simplemente marcarlos como "no activos". • El espacio que recorre cada marciano no es siempre el mismo: el "bloque" formado por todos los marcianos se mueve todo lo que pueda a la derecha y todo lo que pueda a la izquierda. A medida que desaparezcan los marcianos de los extremos, los centrales podrán avanzar más hacia cada lado. – Los marcianos deberán avanzar menos hacia la derecha que si estuvieran solos, por ejemplo con: • Este planteamiento no es fácil calcular cuando se ha matado a un marciano concreto, ni cuando el bloque de marcianos debe avanzar más hacia la derecha o hacia la izquierda. Una opción es crear una "estructura" que representa a cada marciano, con datos como su posición y si está activo: • Y apenas cambia un poco la forma de dibujar a los marcianos, que ahora se hará dentro de un doble bucle "for", para recorrer todos, y comprobando si están activos: O la de comprobar si un disparo ha acertado, con los mismos cambios o hasta donde deben moverse los marcianos, porque habrá que calcular la "primera x" (la del marciano más a la izquierda) y la "última x" (la del marciano más a la derecha): – En base a las condiciones anteriores elabora su algoritmo correspondiente en el lenguaje seleccionado. – Termina el programa **Construir un algoritmo para juego de Marciano 6. Evitar parpadeos- Doble buffer** – Utiliza el lenguaje seleccionado y bibliotecas necesarias. – Emplear doble buffer para evitar parpadeos. El problema es que si dibujamos unas 57 figuras en pantalla cada vez, provoca un serio problema de parpadeo, para corregir evitar parpadeos. – Realiza un "doble buffer" para dibujar en memoria, en vez de hacer directamente en pantalla, y volcar el resultado en el momento final, evitando parpadeos. – En base a las condiciones anteriores elabora su algoritmo correspondiente en el lenguaje seleccionado. – Termina el programa. **Presentación de código** – Presenta las rutinas de programación escritas en editor de texto (código fuente generado) considerando la escritura de rutinas lógicas secuenciales en lenguaje de programación, compuesto por instrucciones en líneas de texto de los programas o algoritmos elaborados, de manera impresa y/o digital. – Compila el programa y generar su código ejecutable, haciendo la corrida, hasta que funcione sin errores.

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
	Realiza el encendido y apagado del equipo de cómputo de acuerdo a indicaciones del fabricante. Entrega un informe de las actividades realizadas en la práctica, formando el portafolio de evidencias.  **ADVERTENCIA DE RIESGO ELÉCTRICO**

II. Guía de Evaluación del Módulo Programación de videojuegos

7. Descripción

La guía de evaluación es un documento que define el proceso de recolección y valoración de las evidencias requeridas por el módulo desarrollado y tiene el propósito de guiar en la evaluación de las competencias adquiridas por los alumnos, asociadas a los Resultados de Aprendizaje; en donde además, describe las técnicas y los instrumentos a utilizar y la ponderación de cada actividad de evaluación. Los Resultados de Aprendizaje se definen tomando como referentes: las **competencias genéricas** que va adquiriendo el alumno para desempeñarse en los ámbitos personal y profesional que le permitan convivir de manera armónica con el medio ambiente y la sociedad; las **disciplinares**, esenciales para que los alumnos puedan desempeñarse eficazmente en diversos ámbitos, desarrolladas en torno a áreas del conocimiento y las **profesionales** que le permitan un desempeño eficiente, autónomo, flexible y responsable de su ejercicio profesional y de actividades laborales específicas, en un entorno cambiante que exige la multifuncionalidad.

La importancia de la evaluación de competencias, bajo un enfoque de **mejora continua**, reside en que es un proceso por medio del cual se obtienen y analizan las evidencias del desempeño de un alumno con base en la guía de evaluación y rúbrica, para emitir un juicio que conduzca a tomar decisiones.

La evaluación de competencias se centra en el desempeño real de los alumnos, soportado por evidencias válidas y confiables frente al referente que es la guía de evaluación, la cual, en el caso de competencias profesionales, está asociada con alguna normalización específica de un sector o área y no en contenidos y/o potencialidades.

El **Modelo de Evaluación** se caracteriza porque es **Confiable** (que aplica el mismo juicio para todos los alumnos), **Integral** (involucra las dimensiones intelectual, social, afectiva, motriz y axiológica), **Participativa** (incluye autoevaluación, coevaluación y heteroevaluación), **Transparente** (congruente con los aprendizajes requeridos por la competencia), **Válida** (las evidencias deben corresponder a la guía de evaluación).

Evaluación de los Aprendizajes.

Durante el proceso de enseñanza - aprendizaje es importante considerar tres finalidades de evaluación: **diagnóstica, formativa y sumativa**.

La evaluación **diagnóstica** nos permite establecer un **punto de partida** fundamentado en la detección de la situación en la que se encuentran nuestros alumnos. Permite también establecer vínculos socio-afectivos entre el docente y su grupo. El alumno a su vez podrá obtener información sobre los

aspectos donde deberá hacer énfasis en su dedicación. El docente podrá **identificar las características del grupo y orientar adecuadamente sus estrategias**. En esta etapa pueden utilizarse mecanismos informales de recopilación de información.

La evaluación **formativa** se realiza durante todo el proceso de aprendizaje del alumno, en forma constante, ya sea al finalizar cada actividad de aprendizaje o en la integración de varias de éstas. Tiene como finalidad **informar a los alumnos de sus avances** con respecto a los aprendizajes que deben alcanzar y advertirle sobre dónde y en qué aspectos tiene debilidades o dificultades para poder regular sus procesos. Aquí se admiten errores, se identifican y se corrigen; es factible trabajar colaborativamente. Asimismo, el docente puede asumir nuevas estrategias que contribuyan a mejorar los resultados del grupo.

Finalmente, la evaluación **sumativa** es adoptada básicamente por una función social, ya que mediante ella se asume una acreditación, una promoción, un fracaso escolar, índices de deserción, etc., a través de **criterios estandarizados y bien definidos**. Las evidencias se elaboran en forma individual, puesto que se está asignando, convencionalmente, un criterio o valor. Manifiesta la síntesis de los logros obtenidos por ciclo o período escolar.

Con respecto al agente o responsable de llevar a cabo la evaluación, se distinguen tres categorías: la **autoevaluación** que se refiere a la valoración que hace el alumno sobre su propia actuación, lo que le permite reconocer sus posibilidades, limitaciones y cambios necesarios para mejorar su aprendizaje. Los roles de evaluador y evaluado coinciden en las mismas personas

La **coevaluación** en la que los alumnos se evalúan mutuamente, es decir, evaluadores y evaluados intercambian su papel alternativamente; los alumnos en conjunto, participan en la valoración de los aprendizajes logrados, ya sea por algunos de sus miembros o del grupo en su conjunto; La coevaluación permite al alumno y al docente:

- Identificar los logros personales y grupales
- Fomentar la participación, reflexión y crítica constructiva ante situaciones de aprendizaje
- Opinar sobre su actuación dentro del grupo
- Desarrollar actitudes que se orienten hacia la integración del grupo
- Mejorar su responsabilidad e identificación con el trabajo
- Emitir juicios valorativos acerca de otros en un ambiente de libertad, compromiso y responsabilidad

La **heteroevaluación** que es el tipo de evaluación que con mayor frecuencia se utiliza, donde el docente es quien, evalúa, su variante externa, se da cuando agentes no integrantes del proceso enseñanza-aprendizaje son los evaluadores, otorgando cierta objetividad por su no implicación.

Actividades de Evaluación

Los programas de estudio están conformados por Unidades de Aprendizaje (UA) que agrupan Resultados de Aprendizaje (RA) vinculados estrechamente y que requieren irse desarrollando paulatinamente. Dado que se establece un resultado, es necesario comprobar que efectivamente éste

se ha alcanzado, de tal suerte que en la descripción de cada unidad se han definido las actividades de evaluación indispensables para evaluar los aprendizajes de cada uno de los RA que conforman las unidades.

Esto no implica que no se puedan desarrollar y evaluar otras actividades planteadas por el docente, pero es importante no confundir con las actividades de aprendizaje que realiza constantemente el alumno para contribuir a que logre su aprendizaje y que, aunque se evalúen con fines formativos, no se registran formalmente en el **Sistema de Administración Escolar SAE**. El **registro formal** procede sólo para las actividades descritas en los programas y planes de evaluación.

De esta manera, cada uno de los RA tiene asignada al menos una actividad de evaluación, a la cual se le ha determinado una ponderación con respecto a la Unidad a la cual pertenece. Ésta a su vez, tiene una ponderación que, sumada con el resto de Unidades, **conforma el 100%.** Es decir, para considerar que se ha adquirido la competencia correspondiente al módulo de que se trate, deberá **ir acumulando** dichos porcentajes a lo largo del período para estar en condiciones de acreditar el mismo. Cada una de estas ponderaciones dependerá de la relevancia que tenga la AE con respecto al RA y éste a su vez, con respecto a la Unidad de Aprendizaje. Estas ponderaciones las asignará el especialista diseñador del programa de estudios.

La ponderación que se asigna en cada una de las actividades queda asimismo establecida en la **Tabla de ponderación**, la cual está desarrollada en una hoja de cálculo que permite, tanto al alumno como al docente, ir observando y calculando los avances en términos de porcentaje, que se van alcanzando (ver apartado 8 de esta guía).

Esta tabla de ponderación contiene los Resultados de Aprendizaje y las Unidades a las cuales pertenecen. Asimismo indica, en la columna de actividades de evaluación, la codificación asignada a ésta desde el programa de estudios y que a su vez queda vinculada al Sistema de Evaluación Escolar SAE. Las columnas de aspectos a evaluar, corresponden al tipo de aprendizaje que se evalúa: **C = conceptual; P = Procedimental y A = Actitudinal**. Las siguientes tres columnas indican, en términos de porcentaje: la primera el **peso específico** asignado desde el programa de estudios para esa actividad; la segunda, **peso logrado**, es el nivel que el alumno alcanzó con base en las evidencias o desempeños demostrados; la tercera, **peso acumulado**, se refiere a la suma de los porcentajes alcanzados en las diversas actividades de evaluación y que deberá acumular a lo largo del ciclo escolar.

Otro elemento que complementa a la matriz de ponderación es la **rúbrica o matriz de valoración**, que establece los **indicadores y criterios** a considerar para evaluar, ya sea un producto, un desempeño o una actitud y la cual se explicará a continuación.

Una matriz de valoración o rúbrica es, como su nombre lo indica, una matriz de doble entrada en la cual se establecen, por un lado, los **indicadores** o aspectos específicos que se deben tomar en cuenta como **mínimo indispensable** para evaluar si se ha logrado el resultado de aprendizaje esperado y, por otro, los criterios o **niveles de calidad o satisfacción alcanzados**. En las celdas centrales se describen los criterios que se van a utilizar para evaluar esos indicadores, explicando cuáles son las características de cada uno.

Los criterios que se han establecido son: **Excelente**, en el cual, además de cumplir con los estándares o requisitos establecidos como necesarios en el logro del producto o desempeño, es propositivo, demuestra iniciativa y creatividad, o que va más allá de lo que se le solicita como mínimo, aportando elementos adicionales en pro del indicador; **Suficiente**, si cumple con los estándares o requisitos establecidos como necesarios para demostrar que se ha desempeñado adecuadamente en la actividad o elaboración del producto. Es en este nivel en el que podemos decir que se ha adquirido la competencia. **Insuficiente**, para cuando no cumple con los estándares o requisitos mínimos establecidos para el desempeño o producto.

Evaluación mediante la matriz de valoración o rúbrica

Un punto medular en esta metodología es que al alumno se le proporcione el **Plan de evaluación**, integrado por la **Tabla de ponderación y las Rúbricas**, con el fin de que pueda conocer qué se le va a solicitar y cuáles serán las características y niveles de calidad que deberá cumplir para demostrar que ha logrado los resultados de aprendizaje esperados. Asimismo, él tiene la posibilidad de autorregular su tiempo y esfuerzo para recuperar los aprendizajes no logrados.

Como se plantea en los programas de estudio, en una **sesión de clase previa a finalizar la unidad**, el docente debe hacer una **sesión de recapitulación** con sus alumnos con el propósito de valorar si se lograron los resultados esperados; con esto se pretende que el alumno tenga la oportunidad, en caso de no lograrlos, de rehacer su evidencia, realizar actividades adicionales o repetir su desempeño nuevamente, con el fin de recuperarse de inmediato y no esperar hasta que finalice el ciclo escolar acumulando deficiencias que lo pudiesen llevar a no lograr finalmente la competencia del módulo y, por ende, no aprobarlo.

La matriz de valoración o rúbrica tiene asignadas a su vez valoraciones para cada indicador a evaluar, con lo que el docente tendrá los elementos para evaluar objetivamente los productos o desempeños de sus alumnos. Dichas valoraciones están también vinculadas al SAE y a la matriz de ponderación. Cabe señalar que **el docente no tendrá que realizar operaciones matemáticas para el registro de los resultados de sus alumnos**, simplemente deberá marcar en cada celda de la rúbrica aquélla que más se acerca a lo que realizó el alumno, ya sea en una hoja de cálculo que emite el SAE o bien, a través de la Web.

8. Tabla de Ponderación

UNIDAD	RA	ACTIVIDAD DE EVALUACIÓN	ASPECTOS A EVALUAR			% Peso Específico	% Peso Logrado	% Peso Acumulado
			C	P	A			
1. Diseño de videojuegos	**1.1** Determina los elementos según el desarrollo de videojuegos con base en tecnologías y herramientas de software.	**1.1.1**	▲		▲	**15**		
	1.2 Elabora el plan de proyecto del desarrollo de videojuegos mediante metodología específica, tareas de ejecución y ayuda de herramientas de software.	**1.2.1**	▲		▲	**20**		
% PESO PARA LA UNIDAD						**35**		
2.Construcción de videojuegos	**2.1** Maneja elementos del entorno de desarrollo para videojuegos mediante herramientas, rutinas con tecnicas de programación y componentes de lenguajes.	**2.1.1**	▲	▲	▲	**25**		
	2.2 Programa aplicaciones de videojuego de acuerdo al diseño establecido y al lenguaje de programación específico.	**2.2.1**	▲	▲	▲	**40**		
% PESO PARA LA UNIDAD						**65**		
PESO TOTAL DEL MÓDULO						**100**		

9. Materiales para el Desarrollo de Actividades de Evaluación

10. Matriz de Valoración o Rúbrica

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema: POVI-02	**Nombre del Módulo:**	**Programación de videojuegos**	**Nombre del Alumno:**	
Docente evaluador:			**Grupo:**	**Fecha:**
Resultado de Aprendizaje:	**1.1** Determina los elementos según el desarrollo de videojuegos con base en tecnologías y herramientas de software.		**Actividad de evaluación:**	**1.1.1** Realiza un cuadro comparativo donde se consideren los elementos dispositivos y herramientas de ambiente de programación para videojuegos.

INDICADORES	%	C R I T E R I O S		
		Excelente	**Suficiente**	**Insuficiente**
ELEMENTOS DE ENTORNO	**30**	Describe lo siguiente: • Entorno de desarrollo de videojuegos • Características de los videojuegos – Funcionamiento en tiempo real. – Realización de tareas independientes al usuario - dibujar los objetos, actualizar coordenadas, calcular colisiones. – Trabajo bajo ciclos en espera de evento- respuesta. • Estructura básica de un videojuego. • Adicionalmente identifica el público hacia donde están dirigidos los videojuegos.	Describe lo siguiente: • Entorno de desarrollo de videojuegos • Características de los videojuegos – Funcionamiento en tiempo real. – Realización de tareas independientes al usuario - dibujar los objetos, actualizar coordenadas, calcular colisiones. – Trabajo bajo ciclos en espera de evento- respuesta. • Estructura básica de un videojuego.	Omite alguno de las siguientes elementos: • Entorno de desarrollo de videojuegos • Características de los videojuegos – Funcionamiento en tiempo real. – Realización de tareas independientes al usuario - dibujar los objetos, actualizar coordenadas, calcular colisiones. – Trabajo bajo ciclos en espera de evento- respuesta. • Estructura básica de un videojuego.
HERRAMIENTAS DE PROGRAMACIÓN	**30**	Describe las siguientes herramientas: • Sistema operativo.	Describe las siguientes herramientas: • Sistema operativo.	Omite alguno de los siguientes elementos: • Sistema operativo.

INDICADORES	%	C R I T E R I O S		
		Excelente	**Suficiente**	**Insuficiente**
		• Lenguajes de programación. • Estructuras y bases de datos. • Editores de texto. • Archivos scripts. • Comunicación de datos a través de redes. • Bibliotecas. • APIs Gráficas. • Ventajas y desventajas de los lenguajes de desarrollo.	• Lenguajes de programación. • Estructuras y bases de datos. • Editores de texto. • Archivos scripts. • Comunicación de datos a través de redes. • Bibliotecas. • APIs Gráficas.	• Lenguajes de programación. • Estructuras y bases de datos. • Editores de texto. • Archivos scripts. • Comunicación de datos a través de redes. • Bibliotecas. • APIs Gráficas.
DISPOSITIVOS PARA VIDEOJUEGOS	**25**	Describe : • Ubicación por generación de cada consola y modelo. • Características de cada consola y tecnologías adicionales que ocupa. • Tipo de dispositivo de almacenamiento y de juego que soporta cada consola y la capacidad en bits. • Ventajas o desventajas del uso de cada dispositivo.	Describe: • Ubicación por generación de cada consola y modelo • Características de cada consola y tecnologías adicionales que ocupa • Tipo de dispositivo de almacenamiento y de juego que soporta cada consola y la capacidad en bits.	Omite ~~la~~ descripción de alguno de los siguientes aspectos**:** • Ubicación por generación de cada consola y modelo. • Características de cada consola y tecnologías adicionales que ocupa. • Tipo de dispositivo de almacenamiento y de juego que soporta cada consola y la capacidad en bits.
PRESENTACIÓN DE LA INFORMACIÓN AUTOEVALUACIÓN	**5**	• Presenta el cuadro comparativo en formato impreso y/o digital (en hoja de cálculo o similar). • Aplica las reglas ortográficas y gramaticales. • Demuestra además orden y limpieza en la información, en forma estructurada.	• Presenta el cuadro comparativo en formato impreso y/o digital. • Aplica las reglas ortográficas y gramaticales.	**Omite alguno de estos aspectos:** • Presenta el cuadro comparativo en formato impreso y/o digital. • Aplica las reglas ortográficas y gramaticales.
ACTITUDES	**10**	• Cumple con asistencia total y participación activa en clase. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar	• Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo.	Carece de lo siguiente: • Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar

INDICADORES	%	C R I T E R I O S		
		Excelente	**Suficiente**	**Insuficiente**
		su trabajo. • Muestra organización y responsabilidad al entregar en fecha previa a la establecida por el docente. • Trabaja con limpieza y orden. • Tiene disposición y asume rol asignado en el trabajo colaborativo	• Muestra responsabilidad al entregar en la fecha establecida por el docente. • Trabaja con limpieza y orden. • Muestra disposición y asume rol asignado en el trabajo colaborativo	su trabajo. • Muestra responsabilidad al entregar en la fecha establecida por el docente. • Trabaja con limpieza y orden. • Muestra disposición y asume rol asignado en el trabajo colaborativo
	100			

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema: POVI-02	**Nombre del Módulo:**	**Programación de videojuegos**	**Nombre del Alumno:**	
Docente evaluador:		**Grupo:**		**Fecha:**
Resultado de Aprendizaje:	**1.2** Elabora el plan de proyecto del desarrollo de videojuegos mediante metodología específica, tareas de ejecución y ayuda de herramientas de software.	**Actividad de evaluación:**	**1.2.1.** Elabora un plan estructural de diseño y acción para el desarrollo de un videojuego	

INDICADORES	**%**	**C R I T E R I O S**		
		Excelente	**Suficiente**	**Insuficiente**
DISEÑO DEL PROYECTO	**50**	Describe: • El asunto en torno al cual se realiza el proyecto y su alcance. • Esquema o bosquejo de la idea en general. • Objetivos. • Tipo de juego a hacer. • Elementos de software necesarios. • Disposición de equipo de cómputo. • Utilización de dibujos, bosquejos, story board de las pantallas. • Comportamiento o mecánica de los objetos del juego. • Estudio de la sintaxis del lenguaje de programación a utilizar. • El lenguaje de desarrollo a ocupar. • Las herramientas de desarrollo para crear las imágenes. • Las características técnicas del equipo de cómputo o dispositivo para el desarrollo. • El equipo de personas de desarrollo o número de programadores. • Las ventajas y limitaciones que se pueden tener con el uso de la	Describe: • El asunto en torno al cual se realiza el proyecto y su alcance. • Esquema o bosquejo de la idea en general. • Objetivos. • Tipo de juego a hacer. • Elementos de software necesarios. • Disposición de equipo de cómputo. • Utilización de dibujos, bosquejos, story board de las pantallas. • Comportamiento o mecánica de los objetos del juego. • Estudio de la sintaxis del lenguaje de programación a utilizar. • El lenguaje de desarrollo a ocupar. • Las herramientas de desarrollo para crear las imágenes. • Las características técnicas del equipo de cómputo o dispositivo para el desarrollo. • El equipo de personas de desarrollo o número de programadores.	Excluye alguno de estos elementos**:** • El asunto en torno al cual se realiza el proyecto y su alcance. • Esquema o bosquejo de la idea en general. • Objetivos. • Tipo de juego a hacer. • Elementos de software necesarios. • Disposición de equipo de cómputo. • Utilización de dibujos, bosquejos, story board de las pantallas. • Comportamiento o mecánica de los objetos del juego. • Estudio de la sintaxis del lenguaje de programación a utilizar. • El lenguaje de desarrollo a ocupar. • Las herramientas de desarrollo para crear las imágenes. • Las características técnicas del equipo de cómputo o dispositivo para el desarrollo. • El equipo de personas de desarrollo o número de programadores.

INDICADORES	%	C R I T E R I O S		
		Excelente	**Suficiente**	**Insuficiente**
		tecnología seleccionada.		
CRONOGRAMA DE ACTIVIDADES	**30**	Define orden, tiempos, holgura y las actividades de: • Diseño del videojuego. • Producción o elaboración de las rutinas de programación. • Prueba piloto y puesta a punto. • Fases de implementación en una gráfica de Gantt elaborada con software de planeación.	Define orden, tiempos, holgura y las actividades de: • Diseño del videojuego. • Producción o elaboración de las rutinas de programación. • Prueba piloto y puesta a punto. • Fases de implementación en una gráfica o tabla elaborada en hoja de cálculo.	Excluye alguno de los siguientes elementos: • Diseño del videojuego. • Elaboración de las rutinas de programación. • Prueba piloto y puesta a punto. • Fases de implementación en una gráfica o tabla elaborada en hoja de cálculo.
PRESENTACIÓN DE LA INFORMACIÓN	**10**	• Presenta el plan de diseño en formato impreso y/o digital, demostrando orden y limpieza, en forma estructurada con un índice e incluyendo las fuentes documentales. • Aplica las reglas ortográficas y gramaticales.	• Presenta el plan de diseño en formato impreso y/o digital, demostrando orden y limpieza. • Aplica reglas ortográficas y gramaticales.	Omite alguno de los siguientes elementos: • Presentación del plan de diseño en formato impreso y/o digital, demostrando orden y limpieza. • Aplicación de las reglas ortográficas y gramaticales.
ACTITUDES	**10**	• Cumple con asistencia total y participación activa en clase. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra organización y responsabilidad al entregar en fecha previa a la establecida por el docente. • Trabaja con limpieza y orden. • Tiene disposición y asume rol asignado en el trabajo colaborativo	• Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra responsabilidad al entregar en la fecha establecida por el docente. • Trabaja con limpieza y orden. • Muestra disposición y asume rol asignado en el trabajo colaborativo	Carece de lo siguiente: • Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra responsabilidad al entregar en la fecha establecida por el docente. • Trabaja con limpieza y orden. • Muestra disposición y asume rol asignado en el trabajo colaborativo
	100			

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema: POVI-02	**Nombre del Módulo:**	**Programación de videojuegos**	**Nombre del Alumno:**	
Docente evaluador:		**Grupo:**		**Fecha:**
Resultado de Aprendizaje:	**2.1** Maneja elementos del entorno de desarrollo para videojuegos mediante herramientas, rutinas con tecnicas de programación y componentes de lenguajes.	**Actividad de evaluación:**	**2.1.1** Realiza rutinas de programación estructurada o *POO* aplicado a casos prácticos de videojuegos con lenguaje de programación seleccionado donde se genere el código fuente y código ejecutable..	

INDICADORES	**%**	**C R I T E R I O S**		
		Excelente	**Suficiente**	**Insuficiente**
MANEJO DE COMPONENTES DE LENGUAJE ESTRUCTURADO	**35**	Elabora rutinas de programación con herramientas del lenguaje (*C++,* Java, C#.) considerando: ▪ El desarrollo de algoritmos de casos prácticos. ▪ La estructura básica de un programa. ▪ Las características generales del lenguaje seleccionado. – Uso de tipos de datos. – Declaración de variables y arreglos. – Operadores principales. – Cambios de estado. – Mostrado y petición de datos. – Estructura de decisión y control. • Además, hace uso de la ayuda desde sitios web en internet.	Elabora rutinas de programación con herramientas del lenguaje (*C++,* Java, C#.) considerando: ▪ El desarrollo de algoritmos de casos prácticos. ▪ La estructura básica de un programa. ▪ Las características generales del lenguaje seleccionado. – Uso de tipos de datos. – Declaración de variables y arreglos. – Operadores principales. – Cambios de estado. – Mostrado y petición de datos. – Estructura de decisión y control.	Omite alguno de los siguientes aspectos: ▪ El desarrollo de algoritmos de casos prácticos. ▪ La estructura básica de un programa. ▪ Las características generales del lenguaje seleccionado. – Uso de tipos de datos. – Declaración de variables y arreglos. – Operadores principales. – Cambios de estado. – Mostrado y petición de datos. – Estructura de decisión y control.
MANEJO DE COMPONENTES DE LENGUAJE POO	**35**	Elabora algoritmos de programación con herramientas del lenguaje POO (*C++,* Java, C#.) considerando: • La estructura básica de un programa y características en	Elabora algoritmos de programación con herramientas del lenguaje POO (*C++,* Java, C#.) considerando: • La estructura básica de un programa y características en POO.	Omite en la elaboración algoritmos de programación POO algún elemento: • La estructura básica de un programa y características en POO.

INDICADORES	%	C R I T E R I O S		
		Excelente	**Suficiente**	**Insuficiente**
		POO. – Clases y objetos. – Propiedades, métodos y eventos. – Encapsulamiento. – Herencia. – Polimorfismo. • Con elementos esenciales de los objetos. • Creando una clase y generando una instancia. • Creando jerarquía de clases por herencia. • Comprobando polimorfismo basado en herencia. • Además, hace uso de la ayuda desde sitios web en internet.	– Clases y objetos. – Propiedades, métodos y eventos. – Encapsulamiento. – Herencia. – Polimorfismo. • Con elementos esenciales de los objetos. • Creando una clase y generando una instancia. • Creando jerarquía de clases por herencia. • Comprobando polimorfismo basado en herencia.	– Clases y objetos. – Propiedades, métodos y eventos. – Encapsulamiento. – Herencia. – Polimorfismo. • Con elementos esenciales de los objetos. • Creando una clase y generando una instancia. • Creando jerarquía de clases por herencia. • Comprobando polimorfismo basado en herencia.
GENERACIÓN DEL CÓDIGO DE PROGRAMACIÓN	**10**	• Instala el compilador, librerías o bibliotecas y estructuras de programación del lenguaje seleccionado (C, C++, Java) • Genera el código ejecutable mediante la compilación de los programas • Verifica que el programa se ejecuta sin errores, en orden y sintaxis del lenguaje de programación. • Realiza pruebas de funcionamiento. • Adiciona comentarios explicativos dentro del desarrollo del código fuente.	• Instala el compilador, librerías o bibliotecas y estructuras de programación del lenguaje seleccionado (C, C++, Java) • Genera el código ejecutable mediante la compilación de los programas • Verifica que el programa se ejecuta sin errores, en orden y sintaxis del lenguaje de programación. • Realiza pruebas de funcionamiento.	Omite alguno de estos elementos: • Instala el compilador, librerías o bibliotecas y estructuras de programación del lenguaje seleccionado (C, C++, Java) • Genera el código ejecutable mediante la compilación de los programas • Verifica que el programa se ejecuta sin errores, en orden y sintaxis del lenguaje de programación. • Realiza pruebas de funcionamiento.
PRESENTACIÓN DE	**10**	Presenta el código fuente, considerando:	Presenta el código fuente, considerando:	Excluye e alguno de los siguientes elementos:

INDICADORES	%	C R I T E R I O S		
		Excelente	**Suficiente**	**Insuficiente**
CÓDIGO		• Escritura de rutinas lógicas secuenciales en lenguaje de programación. • Instrucciones y procedimientos de manera reflexiva durante la programación de componentes. • Instrucciones en líneas de texto de los programas a seguir, de manera impresa y/o digital en el editor del compilador. • Documenta adicionalmente todas las rutinas de programación que va desarrollando.	• Escritura de rutinas lógicas secuenciales en lenguaje de programación. • Instrucciones y procedimientos de manera reflexiva durante la programación de componentes. • Instrucciones en líneas de texto de los programas a seguir, de manera impresa y/o digital en el editor del compilador.	• Escritura de rutinas lógicas secuenciales en lenguaje de programación. • Instrucciones y procedimientos de manera reflexiva durante la programación de componentes. • Instrucciones en líneas de texto de los programas a seguir, de manera impresa y/o digital en el editor del compilador.
ACTITUDES	**10**	• Cumple con asistencia total y participación activa en clase. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra organización y responsabilidad al entregar en fecha previa a la establecida por el docente. • Trabaja con limpieza y orden. • Tiene disposición y asume rol asignado en el trabajo colaborativo	• Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra responsabilidad al entregar en la fecha establecida por el docente. • Trabaja con limpieza y orden. • Muestra disposición y asume rol asignado en el trabajo colaborativo	Carece de lo siguiente: • Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra responsabilidad al entregar en la fecha establecida por el docente. • Trabaja con limpieza y orden. • Muestra disposición y asume rol asignado en el trabajo colaborativo
	100			

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema: POVI-02	**Nombre del Módulo:**	**Programación de videojuegos**	**Nombre del Alumno:**	
Docente evaluador:			**Grupo:**	**Fecha:**
Resultado de Aprendizaje:	**2.2** Programa aplicaciones de videojuego de acuerdo al diseño establecido y al lenguaje de programación específico.		**Actividad de evaluación:**	**2.2.1** Desarrolla aplicaciones de videojuegos donde se genere el código fuente, código ejecutable y pruebas de funcionamiento **HETEROEVALUACIÓN**

INDICADORES	%	C R I T E R I O S		
		Excelente	**Suficiente**	**Insuficiente**
DISEÑO DEL VIDEOJUEGO	**20**	Incluye: • Propuesta, definiendo un curso de acción con pasos específicos. • Objetivo, argumento y mecánica de juego que incluya imágenes o dibujos, bosquejos, story board de las pantallas y efectos de sonido. • Dibujos en modo gráfico de acuerdo con las posibilidades del equipo de cómputo. • Apoyo de herramientas para crear las imágenes (Fractal Design Painter, 3D Studio, Paintbrush, Photoshop, Corel Draw).	Incluye: • Propuesta, definiendo un curso de acción con pasos específicos. • Objetivo, argumento y mecánica de juego que incluya imágenes o dibujos, bosquejos, story board de las pantallas y efectos de sonido. • Dibujos en modo gráfico de acuerdo con las posibilidades del equipo de cómputo.	Omite alguno de los sioguientes elementos**:** • Propuesta, definiendo un curso de acción con pasos específicos. • Objetivo, argumento y mecánica de juego que incluya imágenes o dibujos, bosquejos, story board de las pantallas y efectos de sonido. • Dibujos en modo gráfico de acuerdo con las posibilidades del equipo de cómputo.
GENERACIÓN DEL CÓDIGO DE PROGRAMACIÓN	**25**	• Instala el entorno de desarrollo (compiladores a utilizar, sistema operativo y herramientas alternativas). • Toma como base el compilador, librerías o bibliotecas y estructuras de programación del lenguaje seleccionado (C, C++, Java): • Se apega a la estructura de programas de ciclo de videojuegos (entrada, procesamiento, salida) y	• Instala el entorno de desarrollo (compiladores a utilizar, sistema operativo y herramientas alternativas). • Toma como base el compilador, librerías o bibliotecas y estructuras de programación del lenguaje seleccionado (C, C++, Java): • Se apega a la estructura de programas de ciclo de videojuegos (entrada, procesamiento, salida) y	Omite alguno de estos elementos: • Instala el entorno de desarrollo (compiladores a utilizar, sistema operativo y herramientas alternativas). • Toma como base el compilador, librerías o bibliotecas y estructuras de programación del lenguaje seleccionado (C, C++, Java): • Se apega a la estructura de programas de ciclo de videojuegos

INDICADORES	%	C R I T E R I O S		
		Excelente	**Suficiente**	**Insuficiente**
		finalización. • Genera el código ejecutable mediante la compilación de los programas. • Verifica que el programa se ejecuta sin errores, en orden y sintaxis del lenguaje de programación. • Realiza pruebas de funcionamiento. • Adiciona comentarios explicativos dentro del desarrollo del código fuente.	finalización. • Genera el código ejecutable mediante la compilación de los programas. • Verifica que el programa se ejecuta sin errores, en orden y sintaxis del lenguaje de programación. • Realiza pruebas de funcionamiento. •	(entrada, procesamiento, salida) y finalización. • Genera el código ejecutable mediante la compilación de los programas. • Verifica que el programa se ejecuta sin errores, en orden y sintaxis del lenguaje de programación. • Realiza pruebas de funcionamiento.
DESARROLLO DE LA APLICACIÓN	**30**	Realiza los siguientes procedimientos. • Genera números al azar, evita la espera del teclado con uso de funciones. • Usa "mapas" para memorizar posiciones de la pantalla y obstáculos. • Crea figuras multicolores que se muevan con rutinas. • Evita parpadeos con empleo con uso de doble buffer. • Cambiar la paleta de colores estándar y el tamaño de una barra con arreglos. • Maneja el ratón con bibliotecas de funciones ya creadas. • Reproduce sonidos a través del altavoz o usando bibliotecas de funciones o secuencia de órdenes. • Lee imágenes desde archivos como mapa de bits o vectorial de cualquier formato (JPG, JPEG, GIF, TIFF, BMP, PCX, LBM, PNG,	Realiza los siguientes procedimientos • Genera números al azar, evita la espera del teclado con uso de funciones. • Usa "mapas" para memorizar posiciones de la pantalla y obstáculos. • Crea figuras multicolores que se muevan con rutinas. • Evita parpadeos con empleo con uso de doble buffer. • Cambiar la paleta de colores estándar y el tamaño de una barra con arreglos. • Maneja el ratón con bibliotecas de funciones ya creadas. • Reproduce sonidos a través del altavoz o usando bibliotecas de funciones o secuencia de órdenes. • Lee imágenes desde archivos como mapa de bits o vectorial de cualquier formato (JPG, JPEG, GIF, TIFF, BMP, PCX, LBM, PNG, PSD, CPT). • Carga, muestra, mueve imágenes y	Omite alguno de estos elementos: • Genera números al azar, evita la espera del teclado con uso de funciones. • Usa "mapas" para memorizar posiciones de la pantalla y obstáculos. • Crea figuras multicolores que se muevan con rutinas. • Evita parpadeos con empleo con uso de doble buffer. • Cambiar la paleta de colores estándar y el tamaño de una barra con arreglos. • Maneja el ratón con bibliotecas de funciones ya creadas. • Reproduce sonidos a través del altavoz o usando bibliotecas de funciones o secuencia de órdenes. • Lee imágenes desde archivos como mapa de bits o vectorial de cualquier formato (JPG, JPEG, GIF, TIFF, BMP, PCX, LBM, PNG, PSD, CPT).

INDICADORES	%	C R I T E R I O S		
		Excelente	**Suficiente**	**Insuficiente**
		PSD, CPT). • Carga, muestra, mueve imágenes y avanza columnas creando funciones y órdenes. • Realiza actividades adicionales apoyadas en la programación orientada a objetos.	avanza columnas creando funciones y órdenes.	• Carga, muestra, mueve imágenes y avanza columnas creando funciones y órdenes.
PRESENTACIÓN DE CÓDIGO DE PROGRAMACIÓN	**15**	El código considera: • Escritura de rutinas lógicas secuenciales en lenguaje de programación. • Código fuente y compilación en instrucciones en líneas de texto de los programas a seguir, de manera impresa y/o digital en el editor del compilador. • Documenta adicionalmente todas las rutinas de programación que va desarrollando.	El código considera: • Escritura de rutinas lógicas secuenciales en lenguaje de programación. • Instrucciones en líneas de texto de los programas a seguir, de manera impresa y/o digital en el editor del compilador.	Excluye alguno de estos elementos: • Escritura de rutinas lógicas secuenciales en lenguaje de programación. • Instrucciones en líneas de texto de los programas a seguir, de manera impresa y/o digital en el editor del compilador.
ACTITUDES	**10**	• Cumple con asistencia total y participación activa en clase. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra organización y responsabilidad al entregar en fecha previa a la establecida por el docente. • Trabaja con limpieza y orden. • Tiene disposición y asume rol asignado en el trabajo colaborativo	• Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra responsabilidad al entregar en la fecha establecida por el docente. • Trabaja con limpieza y orden. • Muestra disposición y asume rol asignado en el trabajo colaborativo	Carece de lo siguiente: • Cumple con asistencia al 90 %. • Muestra perseverancia al aprovechar los errores marcados en actividades previas para mejorar su trabajo. • Muestra responsabilidad al entregar en la fecha establecida por el docente. • Trabaja con limpieza y orden. • Muestra disposición y asume rol asignado en el trabajo colaborativo
	100			