|  |  |  |
| --- | --- | --- |
| **Período**  2022-20 | **Asignaturas**  Paradigmas de programación | **Docentes:**  Audo Ali Díaz Gómez |
| **Nombre del Proyecto** | **Juego de memoria.** | |
| **Situación Problema** | En el departamento de Santander, la unidad de emprendimiento de la Universidad Pontificia Bolivariana Seccional Bucaramanga ha diseñado un estudio sobre los beneficios de los juegos de memoria, en especial en niños. Por ello, se contacta con su equipo para desarrollar una solución tecnológica a los siguientes requerimientos.  Se necesita una aplicación que simule el juego de memoria clásico: El juego de memoria es un juego donde las tarjetas tienen imágenes en un lado y cada imagen aparece en dos tarjetas. El juego comienza con todas las cartas boca abajo y los jugadores se turnan para dar la vuelta a dos cartas. Si las dos cartas tienen la misma imagen, se quedan con las cartas, de lo contrario, vuelven a poner las cartas boca abajo. El ganador es la persona con más cartas cuando se han tomado todas las cartas. | |
| **Requisitos del sistema**: | 1. La aplicación debe estar escrita en el lenguaje de programación JAVA. 2. El programa debe contar con una interfaz gráfica que permita a los usuarios jugar con el ratón del computador. 3. El programa debe permitir las modalidades de 1 jugador, 2 jugadores y 1 jugador contra la computadora. 4. El programa debe permitir 3 niveles de dificultad: 4x4, 6x6 y 8x8. 5. El programa debe capturar el nombre del jugador(es) antes de iniciar una nueva partida. 6. El programa debe guardar registro del mayor puntaje de cada jugador en un archivo en formato XML. | |
| **Requisitos para el desarrollo del proyecto:** | * Los equipos deben ser conformados por 2 personas. * La metodología de desarrollo a utilizar es **espiral**, y deberá ser sustentada con evidencia suficiente que valide su implementación. * El desarrollo del proyecto deberá estar enmarcado en el cronograma. * El alcance de cada módulo será validado por el docente y estará en el marco de las competencias del proyecto descrito en este documento. | |
| **Objetivos de Aprendizaje** | * Desarrollar habilidades en el análisis, diseño, desarrollo e implementación de software, contemplando buenos niveles de seguridad, confiabilidad e integridad de los datos. * Desarrollar habilidades de expresión oral. * Desarrollar habilidades para la documentación técnica y para la elaboración de informes. * Desarrollar habilidades para el liderazgo, planeación y ejecución de proyectos. * Decide actuar conforme a los principios, valores, moral y ética profesional que apoyan la confidencialidad, el debido cuidado y el manejo integral de la información. * Diseña e implementa soluciones computacionales con uso de Software Libre para cumplir con los requerimientos establecidos en las fases de especificación con alto nivel de funcionalidad y bajo costo. * Desarrolla interfaces de comunicación de sistemas bajo los estándares y protocolos. * Desarrolla e implementa interfaces de usuario con estética y usabilidad para su visualización. | |
| **Marco Conceptual** | - El Paradigma de la Programación Orientada a Objetos (POO)  \* Objetos y clases: propiedades y comportamientos de las clases.  \* Creación de objetos, métodos constructores.  \* Variables de instancia y variables estáticas.  \* Encapsulamiento y visibilidad para la seguridad de aplicaciones.  \* Creación de métodos.  \* Reutilización: Herencia y polimorfismo.  \* Uso de clases abstractas e interfaces como tipos de datos.  \* Representación gráfica de modelos POO mediante diagramas de clases.  - Entradas y Salidas  \* Modelo de capas de presentación, reglas y modelos.  \* Diseño de formularios de entrada y salida de datos.  \* Validación y conversión de datos de entradas y salida.  \* Flujos y Archivos. E/S estándar. E/S por archivos. | |
| **Habilidades por Evaluar** | * Plantea soluciones a problemas organizacionales, científico y sociales, cumpliendo las etapas del ciclo de vida del software. * Diseña soluciones computacionales con criterios de viabilidad, calidad, seguridad y buenas prácticas, que respondan a requerimientos funcionales y no funcionales. * Desarrolla proyectos de software, aplicando metodologías, enfoques y herramientas de productividad. * Analiza los recursos informáticos requeridos en un proyecto de desarrollo de software. * Emplea técnicas relativas al aseguramiento de la información que cumplan los criterios básicos de la seguridad informática para la protección de los activos de información. | |
| **Fechas y entregables** | |  |  |  | | --- | --- | --- | | **Semana** | **Entregable** | **% Seguimiento** | | Semana 7 (29 agosto – 02 septiembre) | Presentación pre propuesta del proyecto. | 10% Seguimiento corte 1 | | Domingo 04 septiembre | Documento de la propuesta del proyecto. | 20% Seguimiento corte 1 | | Semana 12 (03-07 octubre) | Revisión avance proyecto (con base en cronograma). | 10% Seguimiento corte 2 | | Semana 13 (10-14 octubre) | Pre sustentación. | 10% Seguimiento corte 2 | | Domingo 23 octubre. | Producto. | 30% Seguimiento corte 2 | | Semana 15 (24-28 octubre) | Sustentación. | 10% Seguimiento corte 2 | | Domingo 30 octubre. | Informe final | 10% Seguimiento corte 2 | | |
| **Descripción entregables** | **Presentación pre propuesta del proyecto (5 minutos por grupo)**   * Situación problema * Pregunta de investigación (pregunta problema) * Justificación * Objetivo general * Objetivos específicos   **Documento de la propuesta del proyecto.**   * Situación problema * Pregunta de investigación (pregunta problema) * Justificación * Objetivo general * Objetivos específicos * Metodología (espiral) * Fases – Actividades por cada fase   **Pre sustentación, Sustentación, Informe final:**   * Situación problema * Pregunta de investigación (pregunta problema) * Justificación * Objetivo general * Objetivos específicos * Metodología (espiral) * Fases – Actividades por cada fase * Resultados del proyecto.   + Diagrama de casos de uso.   + Diagrama de clases.   + Pantallas. | |
| **Bibliografía** | Bibliografía básica  [1] G. Booch, J. Rumbaugh, and I. Jacobson, El Lenguaje unificado de modelado. Addison Wesley Longman, 2006.  [2] L. Gervais, Aprender La Programación Orientada A Objetos Con El Lenguaje C#. Eni, 2016.  [3] J. García M, Desarrollo de software dirigido por modelos: conceptos, métodos y herramientas. RA-MA, 2013.  [4] M. Katrib, L. Paneque, L. Leonard, A. Piad, L. Sánchez, and A. Tamayo, Empiece a Programar. Un enfoque multiparadigma con C#. Independently published, 2019.  [5] J. R. Molina Ríos, P. M. Ordóñez, Zea, and J. A. Honores Tapia, Paradigma Orientado a Objetos con Uml: Ingeniería de Software. Vdm Verlag, 2017.  Bibliografía otros idiomas  [6] Ian Griffiths, Programming C# 8.0: Build Cloud, Web, and Desktop Applications. O’Reilly Media, 2020.  [7] J. Skeet, C# in Depth: Fourth Edition. Manning Publications, 2019.  [8] U. Santilal, Functional Programming: Functional Data Structures and Algorithms. Independently published, 2020.  [9] M. Brambilla, J. Cabot, and M. Wimmer, Model-driven software engineering in practice. Morgan & Claypool Publishers, 2017.  [10] E. Lavieri, Mastering Java 11: Develop modular and secure Java applications using concurrency and advanced JDK libraries. Packt Publishing, 2018.  [11] M. Gabbrielli and S. Martini, Programming Languages: Principles and Paradigms. Springer-Verlag, 2010.  [12] N. Ford, Functional Thinking: Paradigm Over Syntax. O’Reilly Media, 2014  Bibliografía en línea  Repositorio Digital UPB: https://bibliotecavirtual.upb.edu.co/pages/index.php | |