**Fase 1: Identificación el Problema y Requerimientos.**

Juego que permita entretener al usuario. El juego se centra el elegir un caballo entre cinco. Apostar una cantidad al caballo y observar si el caballo elegido es el ganador. Solo se le paga al caballo que gane y al segundo.

**Necesidades del usuario:**

1. **Importancia del Ocio:** Las personas necesitan una manera de distraerse. Permitiéndoles disfrutar de un breve tiempo libre jugando a un juego.
2. **Facilidad de Uso y Entretenimiento:** Los usuarios necesitan una interfaz grafica que les permitan interactuar con el juego y observar el transcurso del juego.
3. **Diversidad del Juego**: Se necesita que el juego tenga varios caballos a elegir y que cada caballo pueda ganar. De esta manera, es más entretenido.
4. **Variedad de apuestas:** Es necesario que el usuario pueda elegir la cantidad a apostar por el caballo que eligió.
5. **Recarga de Dinero:** El usuario debe de tener la opción de “recargar” dinero para las apuestas.
6. **Visualización de Resultados:** El usuario debe poder ver claramente que caballo gano, si gano o perdió y cuanto dinero gano o perdió.

**Requerimientos Funcionales:**

1. **Elegir Caballo:**
   * El sistema debe permitir al usuario elegir entre cinco caballos. El usuario debe elegir el caballo antes de que inicie la carrera.
   * El sistema debe verificar que el usuario tenga fondos para la apuesta mínima.
   * Después de elegir el caballo, el sistema debe permitir al usuario elegir su apuesta.
2. **Elegir Apuesta:**
   * El sistema debe permitir al usuario elegir su apuesta. Esta apuesta debe cumplir los siguientes requisitos:
     + Tiene que ser mayor a 50 dólares y debe ser menor o igual a 10.000 dólares.
     + Tiene que ser en múltiplos de 50 dólares, es decir; no puede apostar 70 dólares, por ejemplo.
3. **Cambiar Caballo o Apuesta:**
   * Antes del inicio de la carrera, se debe permitir al usuario cambiar el caballo, la apuesta o ambos.
4. **Carrera:**
   * Antes de iniciar la carrera, se informa al usuario del caballo elegido y la apuesta realizada, pidiendo una confirmación para iniciar la carrera. Si el usuario confirma, la carrera inicia, por otro lado, si no confirma, permite al usuario cambiar el caballo o la apuesta.
   * Cuando la carrera termine, se debe informar al usuario que caballo gano, y cuánto dinero gano o perdió.
5. **Pago Carrera:**
   * Cuando se acabe la carrera, se revisa la posición del caballo por el cual se aposto:
     + Si el caballo elegido queda de primero, este caballo gana. Se paga lo apostado por 4.7. Es decir, si se aposto la mínima (50 dólares), se hace el siguiente calculo: . El jugador gana en total 235 dólares.
     + Si el caballo elegido queda en segunda posición, colocado. Se paga lo apostado por . Es decir, si se aposta la mínima (50 dólares), se hace el cálculo del ganador, dando como resultado 235 dólares. A este resultado, se le resta la apuesta, . Este valor, se divide entre tres y se le suma la apuesta realizada: . Se realiza la función de suelo, es decir, se quita el decimal y se aproxima el valor al múltiplo de 10 más cercano. . El jugador gana en total 110 dólares.
     + Si el caballo elegido no queda de primero ni de segundo. El jugador pierde el dinero apostado.
6. **Recargar Dinero:**
   * El sistema debe permitir al usuario realizar una recarga de dinero. Esta recarga debe:
     + Ser mínimo de 250 dólares y de un máximo de 100.000 dólares.
     + Ser de múltiplos de 10 dólares, es decir, no puede recargar 122 dólares, por ejemplo.
   * Para recargar dinero, el usuario debe no estar participando en una carrera.
7. **Visualizar Resultados Anteriores:**
   * El sistema debe informar al usuario del resultado de todas las carreras en las que ha participado. Se informa a través de la cantidad de carreras que ha ganado cada caballo y a través del porcentaje de victoria que tiene cada caballo. Es decir, caballo “x” ha ganado 4 carreras, ha ganado el 20% de todas las carreras.
8. **Permitir Cambiar Algoritmo:**
   * El sistema debe permitir al usuario cambiar el algoritmo que se va a utilizar en la carrera. Cambiar el algoritmo de Dijkstra al algoritmo de Floyd-Warshall y viceversa.

**Fase 2: Investigación.**

**Definiciones:**

1. **¿Cómo se gana?:**

Se gana eligiendo el caballo que quede en primera posición o se gana un dividendo si el caballo elegido queda de segundo. La puesta sobre el caballo ganador se paga por 4.7. La apuesta por el segundo es la cuota colocada de .

1. **¿Como se calcula que caballo va a ganar?:**

Este juego cuenta con un algoritmo aleatorio realizado con base en teoría de grafos. De esta manera el caballo que gana es completamente aleatorio. Sin embargo, se muestran los últimos ganadores, por si el usuario se quiere basar en las probabilidades que ha tenido cada caballo en ganar.

1. **Grafo:**

Un grafo es un conjunto de objetos que tienen pueden tener relación entre ellos. Un grafo está formado por vértices y por aristas que conectas los vértices entre ellos. Estas aristas pueden ser dirigidas o no dirigidas. A la vez, cada arista puede tener un peso o no tener peso.

Los grafos dirigidos se caracterizan por tener aristas asociadas a una dirección (las aristas no son equivalentes). Por otro lado, los grafos no dirigidos tienen la característica que las aristas son equivalentes .

En este caso, se va a utilizar un grafo que la principal característica es que las aristas tienen un peso.

1. **Pista como grafo:**

Algunos juegos de carreras tienen un sistema de “checkpoint”. Esto significa que la pista se divide en distintos puntos de control. Dando como resultado que el jugador debe pasar por todos los puntos de control antes de pasar por la meta para que la vuelta cuente. Un ejemplo para entender el concepto es el circuito hamiltoniano; es un circuito que visita cada vértice una vez sin repeticiones, este inicia y termina en el mismo vértice.

Con estos conceptos se genero la idea de crear una pista que inicie en un vértice, y pueda pasar por distintos vértices, que tienen peso, que conforman la pista, llegando a un vértice final.

1. **Algoritmos de Dijkstra:**

El Algoritmo de Dijkstra comienza desde el nodo de origen y explora el grafo en busca del camino más corto hacia todos los demás nodos.

El algoritmo realiza un seguimiento de la distancia más corta conocida hasta el momento entre el nodo de origen y cada otro nodo, actualizando estos valores si se descubre un camino más corto.

Una vez que el algoritmo ha identificado el camino más corto entre el nodo de origen y otro nodo, ese nodo se etiqueta como "visitado" y se incorpora al camino.

Este proceso continúa hasta que se hayan visitado todos los nodos del grafo. Como resultado, obtenemos un camino que conecta el nodo de origen con todos los demás nodos, siguiendo la ruta más corta disponible para alcanzar cada nodo.

1. **Algoritmo de Floyd-Warshall:**

El algoritmo de Floyd-Warshall, se centra en encontrar el camino mas corto entre todos los vértices en un grafo con peso. Este algoritmo funciona para grafos dirigidos y no dirigidos.

Funciona mediante la construcción de una matriz que contiene las distancias más cortas entre los nodos. Luego, compara todas las rutas posibles a través de todos los nodos para actualizar la matriz con los caminos más cortos. Esto se repite hasta que se obtenga la matriz final de distancias más cortas. El algoritmo es eficiente para grafos pequeños y funciona incluso cuando hay ciclos negativos en el grafo.

**Fase 3: Búsqueda de ideas creativas. http://www.ijmlc.org/papers/82-A1090.pdf**

**Lluvia de ideas:**

1. **Grafos como pista:** Crear una pista que inicie en un vértice, y pueda pasar por distintos vértices que estén conectados por aristas y que cada arista tenga un peso. Dando como resultado, que la pista inicie en un vértice y con distintos algoritmos se encuentre el camino hasta el último vértice.

Fase 4: Transición de la formulación de ideas de diseños preliminares.

Ideas:

1. IA de auto clasificación: Si bien es una idea atractiva, presenta diversos problemas que rozan con la propia concepción de esta idea. Por ejemplo, los usuarios podrían preferir ser quienes determinan la prioridad de sus tareas para reafirmar su capacidad de toma de decisiones y de su libertad, en lugar de delegarle algo tan importante como su lista de acciones a una IA.

También podría sucedes que si el usuario no es suficientemente especifico y enfático con alguna tarea, la IA se confundiría con la semántica y determinaría un orden de decisión que desfavorecería al usuario. Esto podría generar múltiples quejas e incluso demandas, por lo cual resulta poco práctica.

Además, el tiempo y costo de desarrollo e implementación de esta IA, podría ser demasiado para una tarea que puede ser tan sencilla en comparación con otras. Lo cual la vuelve poco rentable.

1. El gestor de tareas tipo bloc de notas: Es una idea que presenta múltiples beneficios asociados, pues a pesar de que la sencillez pueda hacerla muy regular para algunos usuarios, esto mismo podría ser lo que la haga destacar. Pues ser sencilla y muy intuitiva, podría cumplir su función de hacer la vida del usuario más fácil. Bastaría con mostrarle al usuario la información y tareas que él determinó en su propio ejercicio de decisión. Y la app ser una forma sencilla de que el usuario recuerde su tareas como lo haría con una libreta.
2. Sistema de recompensas: Este sistema podría incentivar enormemente al usuario a cumplir con sus labores. No obstante, es complejo pensar en las posibles recompensas al usuario, además de que en general, el sistema no generará ningún beneficio a partir del usuario. Es por esto que si el sistema no genera algo a favor, involucra alguna competencia o algo por el estilo, no podría brindar una recompensa. A parte de eso, se podría decir que así se pierde la visión del objetivo del sistema, pues se pretende darle a la gente una herramienta de gestión y recordatorios para poder acomodarse mejor, y no tener que almacenar todo el tiempo sus tareas en la cabeza, sino que mejor dejarlo en la memoria de sus dispositivos lista para ser consultada.
3. Notificaciones creativas: Si bien es una solución realmente creativa y alternativa, podría ser molesta para algunos usuarios, molestar a los usuarios con algún tipo de autismo o estrés postraumático. El tiempo de desarrollo también es un factor en contra, sin mencionar que sería necesario acomodar el papeleo legal para poder usar las voces de famosos en la app.
4. Sistema de vinculación emocional: Esta revolucionaria propuesta ayudaría a los usuarios a desarrollar un mejor compromiso con sus tareas y pendientes. Usar una interfaz que involucre un avatar podría ayudar a la interacción con el sistema y a la familiaridad con este. No obstante, la forma en la que el avatar puede llegar a quejarse o morir, puede resultar en extremo molesta, insensible y cruel para algunos usuarios, por lo cual se vuelve poco práctica.
5. Sistema de notificaciones vergonzosas: La propuesta parte de una considerable y prudente base teórica, pero el constante ejercicio de ser expuesto a notificaciones aleatorias vergonzosas podría afectar al usuario en su vida privada, además de que podrían sonar en el peor momento. Es por esto que a pesar de que en una primera revisión, esta pueda parecer prometedora, alternativa y completamente viable, la realidad es que al detallar el funcionamiento y como se relacionaría en un entorno con una persona real, posiblemente al comienzo sea una app divertida, pero al poco tiempo será una app molesta a desinstalar.

**Paso 5. Evaluación y Selección de la Mejor Solución.**

Criterios:

1. Facilidad en implementación:

Fácil de implementar: 1

Complejidad regular: 2

Difícil de implementar: 3

1. Facilidad de Uso:

Fácil de usar y muy intuitiva: 1

Sencilla: 2

Uso regular: 3

Complicada: 4

Complicada y poco intuitiva: 5

1. Personalización:

Muy personalizable: 1

Poco personalizable: 2

Nada personalizable: 3

1. Costo:

Poco costosa: 1

Normal: 2

Costosa: 3

1. Creatividad:

Muy creativa: 1

Creativa: 2

Poco creativa: 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Solución | Inteligencia Artificial para Priorización | Gestor de tareas tipo Bloc de notas | Sistema de Recompensas | Sistema interactivo de notificaciones creativas | Sistema de vinculación emocional del usuario y sus tareas | Sistema de notificaciones vergonzosas |
| Criterios |
| Facilidad de implementación | 3 | 1 | 2 | 3 | 3 | 3 |
| Facilidad de uso | 3 | 1 | 4 | 4 | 5 | 4 |
| Personalización | 2 | 3 | 2 | 1 | 1 | 1 |
| Costo | 3 | 1 | 3 | 3 | 3 | 3 |
| Creatividad | 2 | 2 | 2 | 1 | 1 | 1 |
| Total | 13 | 8 | 11 | 12 | 13 | 12 |

Como los criterios son 1 para el mejor y 3 para el peor, entonces el total más bajo es el mejor. Esto nos arroja que el gestor de tareas tipo bloc de notas es la mejor solución.