1. **Contexto de pruebas.**

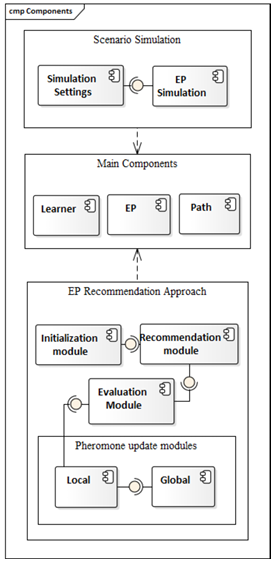
*1.1* *Proyecto / Pruebas / tipos de pruebas.*

El algoritmo *ACO4MLPR* (*Ant Colony Optimization for Microlearning Path Recommendation*) es un enfoque basado en la optimización por colonia/s de hormigas. Su objetivo principal es generar rutas de microaprendizaje adaptadas a las necesidades individuales de los aprendices, optimizando los costos asociados a las transiciones entre píldoras educativas (PE) y ajustándose al rendimiento y preferencias del aprendiz.

El objetivo de las pruebas es validar el correcto funcionamiento, estabilidad y eficiencia del algoritmo ACO4MLPR bajo diferentes escenarios y configuraciones. Estas pruebas se limitan a la observación y documentación de los resultados generados por el algoritmo, ya que no se realizará corrección de defectos o cambios en el código fuente.

*1.2* *Objetos de pruebas.*

Los objetos de prueba son los componentes específicos del algoritmo ACO4MLPR que serán evaluados durante las pruebas: (Es necesario actualizar el diagrama de componentes agregando los módulos de Concurrencia y GUI)



* *Módulo de Inicialización.*

Configura los valores iniciales del algoritmo, como las feromonas iniciales, los niveles de conocimiento de los aprendices y las características de las píldoras educativas.

* *Módulo de Recomendación.*

Encargado de sugerir píldoras educativas (PE) al aprendiz basándose en las probabilidades calculadas con las feromonas acumuladas y la visibilidad heurística.

* *Módulo de Evaluación.*

Responsable de evaluar el desempeño del aprendiz en cada píldora educativa (PE) y determinar su progreso en función de sus calificaciones.

* *Módulo de Actualización Local de Feromonas.*

Ajusta las feromonas de un camino inmediatamente después de que un aprendiz completa una píldora educativa.

* *Módulo de Actualización Global de Feromonas.*

Realiza ajustes de feromonas al final de un recorrido completo del aprendiz, considerando el costo total de la ruta.

* *(Discutir) Módulo de Concurrencia.*

Implementación que permite a múltiples aprendices interactuar simultáneamente con el sistema mediante hilos (threads).

* *(Discutir) Interfaz Gráfica de Configuración.*

Herramienta utilizada para configurar los parámetros experimentales y ejecutar simulaciones del algoritmo, proporcionando una interfaz amigable para los usuarios.

*1.3* *Alcance de las pruebas.*

Se probará solo los módulos que conforman al algoritmo ACO4MLPR, descritos en el apartado 1.2.

Se realizarán pruebas A/B entre el ACO4MLPR y un algoritmo ACO tradicional para microaprendizaje, con el objetivo de comparar el desempeño del ACO4MLPR sobre otros algoritmos tradicionales.

Las pruebas no incluirán corrección de defectos o cambios en el código fuente del algoritmo. Cualquier hallazgo será documentado únicamente para análisis posterior.

*1.4* *Bases de pruebas.*

Las bases de pruebas incluyen los siguientes documentos para diseñar las pruebas:

* *Algoritmo bioinspirado aplicado a la recomendación de rutas de microaprendizaje en educación superior.*
* *ISO/IEC/IEEE 29119-1:2022*
* *ISO/IEC/IEEE 29119-3:2021*
* *ISO/IEC/IEEE 29119-4:2021*

1. **Suposiciones y limitaciones.**

Suposiciones:

* El algoritmo ACO4MLPR está correctamente implementado según la documentación técnica provista.
* Los datos de entrada simulados (perfiles de aprendices, características de las píldoras educativas) son representativos de un entorno real.
* Las pruebas comparativas con el algoritmo ACO tradicional usan configuraciones equivalentes para garantizar una evaluación justa.
* No se realizarán modificaciones en el código fuente del algoritmo durante la ejecución de las pruebas.

Limitaciones:

* No se evalúa el rendimiento del algoritmo en producción, solo en entornos simulados.
* Las pruebas no evalúan aspectos relacionados con la interfaz gráfica de configuración en profundidad, más allá de su capacidad para configurar parámetros experimentales.
* El enfoque principal de las pruebas está en la comparación funcional y de desempeño entre el ACO4MLPR y el ACO tradicional, no en la optimización del algoritmo.

1. **Stakeholders.**

Las partes interesadas en el proceso de pruebas y sus respectivos roles, en este caso yo opino que deberían de ser el Dr. Saul y la Dr. Alma, ya que el algoritmo es para ver si se puede implementar en un sistema más completo. (Profesores que requieren de una capacitación)

1. **Registro de riesgos.**

Para el registro de los riesgos se toma en cuenta la siguiente terminología:

* *P (Probabilidad):* Indica la posibilidad de que el riesgo ocurra. Se evalúa en una escala del 1 al 6, donde 6 es la mayor probabilidad de ocurrencia.
* *C (Consecuencia o Impacto):* Representa el efecto que tendría el riesgo si se materializa. También se mide en una escala del 1 al 6, donde 6 representa el mayor impacto.
* *E (Exposición):* Es el producto de P × C, lo que proporciona una medida del nivel de riesgo asociado a cada situación. Los valores altos de E indican riesgos que requieren mayor atención y medidas de mitigación.

*4.1* *Riesgos del producto.*

Riesgos relacionados con posibles defectos en el producto, como problemas en funcionalidad, rendimiento o seguridad.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Riesgo** | **P** | **C** | **E** | **Actividades de Mitigación** |
| RD1 | Rutas de aprendizaje no óptimas o inconsistentes | 5 | 6 | 30 | Revisión exhaustiva de los casos de prueba funcionales y pruebas exploratorias para identificar inconsistencias en las rutas generadas. |
| RD2 | Alto tiempo de ejecución en pruebas de carga | 4 | 5 | 20 | Optimización de parámetros de rendimiento y ejecución de pruebas de estrés para validar la escalabilidad. |
| RD3 | Impacto negativo de la actualización local de feromonas en escenarios con pocos aprendices | 3 | 5 | 15 | Ajuste de parámetros en pruebas de configuración y ejecución de pruebas A/B para comparar con el enfoque tradicional. |
| RD4 | Fallo en la personalización de rutas según las características del aprendiz | 3 | 6 | 24 | Verificar que el módulo de recomendación use correctamente los datos de los aprendices y ajuste adecuadamente la ruta. |
| RD5 | Degradación incorrecta de niveles de aprendizaje | 3 | 5 | 15 | Validar mediante pruebas funcionales que el sistema promueve o degrada correctamente a los aprendices en función de sus calificaciones. |
| RD6 | Fallos en la integración de los módulos de evaluación y recomendación | 4 | 5 | 20 | Pruebas de integración enfocadas en la transferencia de información correcta entre módulos. |

*4.2* *Riesgos del proyecto.*

Riesgos asociados a la ejecución del proyecto de pruebas, como retrasos en el cronograma o falta de recursos.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Riesgo | P | C | E | Actividades de Mitigación |
| RY1 | Falta de recursos computacionales para pruebas de alto volumen | 3 | 4 | 12 | Planificación anticipada de recursos y optimización de entornos virtuales para pruebas concurrentes. |
| RY2 | Cambios en los requerimientos del algoritmo durante la fase de pruebas | 5 | 5 | 25 | Documentación estricta de requerimientos y comunicación continua con los stakeholders. |
| RY3 | Limitaciones en el tiempo disponible para pruebas. | 4 | 4 | 16 | Priorización de casos de prueba críticos y ¿automatización de pruebas repetitivas? |
| RY4 | Dependencia de la disponibilidad de infraestructura y configuración en Docker | 3 | 4 | 12 | Verificación anticipada del entorno de pruebas y ajustes de configuración previos a la ejecución. |
| RY5 | Modificaciones necesarias en futuras versiones del algoritmo tras los hallazgos de pruebas | 4 | 5 | 20 | Documentación detallada de defectos y análisis de impacto en futuras iteraciones del proyecto. |

1. **Estrategia de pruebas.**

*5.1* *Nivel de pruebas.*

Los niveles de prueba aplicados son:

|  |  |  |
| --- | --- | --- |
| Nivel de Prueba | Descripción | Módulos Probados |
| Pruebas Unitarias | Validar funciones individuales y cálculos específicos. | * Módulo de Inicialización. * Módulo de Recomendación. * Módulo de Evaluación. * Módulo de Actualización de Feromonas. |
| Pruebas de Integración | Verificar la comunicación y el flujo de información entre módulos. | * Módulo de Recomendación. * Módulo de Evaluación. * Módulo de Actualización de Feromonas. * Módulo de Concurrencia. |
| Pruebas Funcionales | Validar que el sistema cumple con los requerimientos funcionales. | * Todos los módulos |
| Pruebas de Rendimiento | Evaluar el tiempo de ejecución y escalabilidad del sistema. | * Módulo de Recomendación * Módulo de Concurrencia |

Pruebas de sistema

(Faltan especificar Diagramas de estado o las clases del mismo código fuente)

Las pruebas serán ejecutadas en el siguiente orden:

1. **Pruebas Unitarias:** Se validan los cálculos básicos y la inicialización de parámetros en módulos individuales.
2. **Pruebas de Integración:** Se verifica la interacción entre módulos clave para asegurar que la comunicación fluya correctamente.
3. **Pruebas Funcionales:** Se evalúa el sistema en su conjunto para confirmar que cumple con sus objetivos y requerimientos.
4. **Pruebas de Rendimiento:** Se analiza la capacidad del sistema para manejar diferentes volúmenes de aprendices y configuraciones.
5. **Pruebas A/B:** Se comparan los resultados de ACO4MLPR con un algoritmo ACO tradicional para validar mejoras y diferencias.

*5.2* *Tipos de pruebas.*

|  |  |
| --- | --- |
| Tipo de Prueba | Descripción |
| Pruebas Unitarias | Validan el correcto funcionamiento de las funciones individuales del algoritmo. |
| Pruebas de Integración | Verifican la correcta interacción entre los módulos que componen el sistema. |
| Pruebas Funcionales | Evalúan si el sistema cumple con los requerimientos funcionales definidos. |
| Pruebas de Rendimiento | Analizan el tiempo de ejecución y escalabilidad del algoritmo en diferentes condiciones de carga. |
| Pruebas de Recuperación | Evalúan el comportamiento del sistema frente a datos inconsistentes o entradas no válidas. |
| Pruebas de Estrés | Determinan la capacidad del sistema en condiciones de uso extremo. |
| Pruebas Exploratorias | Exploran configuraciones y escenarios inesperados para detectar fallos no previstos. |
| Pruebas A/B | Comparan el desempeño del ACO4MLPR con un ACO tradicional para validar mejoras en eficiencia y personalización. |

Pruebas que NO se realizarán:

* *Pruebas de Seguridad:* No se validarán aspectos de seguridad, ya que el enfoque del plan de pruebas se centra en las funcionalidades del algoritmo.
* *Pruebas de Regresión y Retesting:* No se realizarán estas pruebas, ya que no se corregirán defectos o se modificarán partes del código base.

*5.3* *Entregables de pruebas.*

Los siguientes documentos se generarán como resultado del plan de pruebas:

|  |  |
| --- | --- |
| Documento | Descripción |
| Plan de Pruebas | Define el alcance, estrategias, niveles y tipos de pruebas a realizar sobre el algoritmo ACO4MLPR. |
| Especificación de Casos de pruebas | Define los casos de pruebas identificados para el plan de prueba. |
| Bitácora de Pruebas | Registro de la ejecución de pruebas, documentando el estado de cada caso y sus resultados. |

*5.4* *Técnicas de diseño de pruebas.*

Las siguientes técnicas de diseño de pruebas se utilizarán para la generación de los casos de pruebas que se ejecutarán sobre el ACO4MLPR:

|  |  |
| --- | --- |
| Técnica de Diseño de Pruebas | Descripción |
| Particiones de equivalencia. | Divide los datos de entrada en clases equivalentes para reducir el número de casos de prueba necesarios. |
| Análisis de valores límite (BVA). | Prueba los valores en los límites de los rangos de entrada para detectar errores en extremos críticos. |
| Tabla de decisiones. | Evalúa combinaciones complejas de reglas de negocio mediante tablas de decisión. |
| Transición de estados. | Verifica el comportamiento del sistema cuando cambia de un estado a otro. |
| Árbol de clasificación. | Organiza los datos de prueba en una estructura de árbol para cubrir diferentes combinaciones de entrada. |
| Pruebas combinatorias. | Aplica técnicas como el análisis de pares para optimizar la cantidad de combinaciones de datos de prueba. |
| Prueba de casos de uso. | Prueba el sistema según escenarios reales de uso basados en casos de uso. |
| Pruebas exploratorias. | Se enfoca en la búsqueda de errores sin casos de prueba predefinidos, explorando la funcionalidad del sistema. |
| Pruebas de carga. | Evalúa cómo se comporta el sistema bajo una carga esperada de usuarios y solicitudes. |
| Pruebas de estrés. | Examina el rendimiento del sistema bajo condiciones extremas de carga y uso. |

*5.5* *Criterios de entrada y salida.*

Establece los criterios para iniciar y finalizar las pruebas, asegurando que se cumplan las condiciones necesarias.

*Criterios de Entrada*: Las pruebas de ACO4MLPR iniciarán únicamente cuando se cumplan las siguientes condiciones:

1. El plan de pruebas ha sido aprobado.
2. Los casos de prueba han sido diseñados y revisados.
3. El entorno de pruebas ha sido configurado correctamente (contenedor Docker con el algoritmo).
4. Se cuenta con los datos de prueba necesarios para ejecutar los casos definidos.

*Criterios de Salida*: Las pruebas finalizarán cuando se cumplan los siguientes criterios:

1. Se ha ejecutado el 100% de los casos de prueba definidos.
2. No hay fallos críticos pendientes de resolución.
3. Se han documentado todos los resultados en la Bitácora de Pruebas.
4. Se ha generado el reporte final con los hallazgos y conclusiones de la prueba.

*5.6* *Criterio de finalización de pruebas.*

Los siguientes criterios establecen cuando un caso de prueba se encuentra completado:

1. Se ha ejecutado al menos el 95% de los casos de prueba planificados, considerando que algunos pueden quedar bloqueados por restricciones externas.
2. No quedan defectos críticos abiertos que impidan la operatividad del sistema.
3. Los resultados de las pruebas han sido documentados y analizados en la Bitácora de Pruebas.
4. Se ha realizado la comparación con un ACO tradicional en las pruebas A/B y se han registrado los hallazgos.
5. Se ha completado el informe final de pruebas con el resumen de los hallazgos y el análisis de desempeño del algoritmo.
6. Se han alcanzado los objetivos de prueba definidos en el Plan de Pruebas.

*5.7* *Nivel de independencia.*

El nivel de independencia de las pruebas en este plan es el siguiente:

* Pruebas Unitarias: Serán realizadas por el tester principal, el cual será el que diseña y ejecuta las pruebas, ya que requieren conocimiento detallado del código y la lógica del algoritmo.
* Pruebas de Integración: Se llevarán a cabo por el tester principal, asegurando que los módulos interactúan correctamente entre sí.
* Pruebas Funcionales y de Rendimiento: Serán realizadas por el tester principal, quien analizará el comportamiento del sistema sin intervención de los desarrolladores.
* Pruebas A/B: El tester principal se encargará de comparar los resultados obtenidos por ACO4MLPR con un ACO tradicional, garantizando un análisis imparcial.

*5.8* *Métricas a coleccionar.*

*Métricas Generales (aplicables a todos los tipos de prueba):*

* Número total de casos de prueba ejecutados.
* Número de casos de prueba exitosos.
* Número de fallos detectados en la ejecución de pruebas.
* Tasa de éxito de las pruebas = (Casos exitosos / Total de casos ejecutados) \* 100.

*Métricas de específicas por tipo de prueba:*

|  |  |
| --- | --- |
| Tipo de Prueba | Métricas Específicas |
| Pruebas Unitarias | * Número de funciones probadas. * Número de errores detectados en cada módulo. |
| Pruebas de Integración | * Número de errores de comunicación entre módulos. * Tiempo de respuesta en la interacción entre módulos. |
| Pruebas Funcionales | * Precisión de las rutas generadas (comparando con rutas óptimas esperadas). * Número de iteraciones necesarias para converger a una solución estable. |
| Pruebas de Rendimiento | * Tiempo de ejecución del algoritmo en diferentes escenarios. |
| Pruebas A/B | * Comparación de costos de ruta entre ACO4MLPR y ACO tradicional. |
| Pruebas de Estrés y Resistencia | * Porcentaje de ejecuciones que fallan bajo cargas extremas. |

*5.9* *Requerimientos de los datos de prueba.*

Especifica los datos necesarios para ejecutar las pruebas, asegurando que estos sean representativos y válidos.

Los datos de prueba serán generados de manera sintética para simular perfiles de aprendices con distintas características y niveles de conocimiento.

Se crearán conjuntos de datos con diferentes cantidades de aprendices para evaluar la escalabilidad del algoritmo (pequeña, mediana y grande escala).

Los datos serán almacenados en archivos estructurados (CSV o JSON) para facilitar su uso en las pruebas.

Dado que los datos serán generados artificialmente, no se requiere control de datos sensibles.

Se garantizará que los conjuntos de datos utilizados en pruebas puedan ser replicados en diferentes ejecuciones para asegurar la validez de los resultados.

Características de los Datos:

* Perfiles de aprendices con diferentes niveles de conocimiento (básico, intermedio, avanzado).
* Preferencias de contenido educativo.
* Evaluaciones simuladas con calificaciones en el rango de 0 a 100.
* Costos de transición entre píldoras educativas, modelados de manera realista.

*5.10* *Requerimientos del entorno de prueba.*

(Falta)

*5.11* *Criterio de suspensión y reanudación.*

Los siguientes criterios de suspensión y reanudación de pruebas establecen las condiciones bajo las cuales las pruebas pueden ser temporalmente detenidas y las condiciones que deben cumplirse para retomarlas.

*Criterios de Suspensión:*

1. Se detectan fallos críticos en el algoritmo que impiden continuar con las pruebas (ejemplo: bucles infinitos, errores en la generación de rutas, tiempos de respuesta excesivos).
2. Se presentan problemas en los datos de prueba que afectan la validez de los resultados.
3. El entorno de pruebas sufre fallos técnicos que impiden la ejecución de los casos de prueba.
4. Se identifican inconsistencias graves en los resultados que requieren revisión y validación antes de continuar.

*Criterios de Reanudación:*

1. Se han corregido o mitigado los fallos críticos que causaron la suspensión.
2. Se han validado y restablecido los datos de prueba para asegurar su integridad.
3. El entorno de pruebas está nuevamente operativo y estable.
4. Se han evaluado y documentado las inconsistencias detectadas, asegurando que no afectan la validez del resto de las pruebas.

*Autoridad para Suspender y Reanudar Pruebas:* (Discutir)

*5.12* *Casos de prueba.*

(Documentos Aparte)

|  |  |
| --- | --- |
| ID | CP-001 |
| Titulo | Validar que los valores por defecto se cargan correctamente si no se proporcionan parámetros personalizados. | | | | |
| Prioridad | 3 | Objeto de Prueba | Módulo de Inicialización | Estado | Pendiente de ejecución. |
| Precondiciones | | El sistema debe estar en un estado de configuración inicial sin valores personalizados. | | | |
| Entradas | | No se proporcionan parámetros de configuración. | | | |
| Resultado esperado | | Los valores iniciales deben establecerse correctamente según los valores por defecto definidos en el código fuente. | | | |
| Postcondición | | El sistema queda en un estado listo para ejecutar pruebas adicionales. | | | |

*5.13* *Proceso de prueba.*

(Definir y especificar más la plantilla de especificación de proceso de pruebas.)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID Procedimiento de Prueba | Objetivo y prioridad | | | | Duración estimada |
|  |  | | | |  |
| Inicio: | | | | | |
| Relación con otros procedimientos: | | | | | |
| Registro de Prueba | | | | | |
| Fecha | Tester | | Objeto de Prueba | | Estado |
|  |  | |  | |  |
| Comentarios: | | | | | |
| Procedimientos | | | | | |
| ID Caso de Prueba | Actividades | Examinación del resultado | | Resultados reales. | Resultados del Caso de Prueba |
|  |  |  | |  |  |
|  |  |  | |  |  |
| Finalización: | | | | | |

1. **Cronograma de actividades y recursos.**

*6.1 Cronograma.*

Detalla las actividades específicas de prueba y sus estimaciones de tiempo y recursos.

*6.2 Recursos humanos.*

Describe los recursos humanos necesarios, incluyendo la cantidad de personal, sus habilidades y el tiempo que dedicarán a las pruebas.