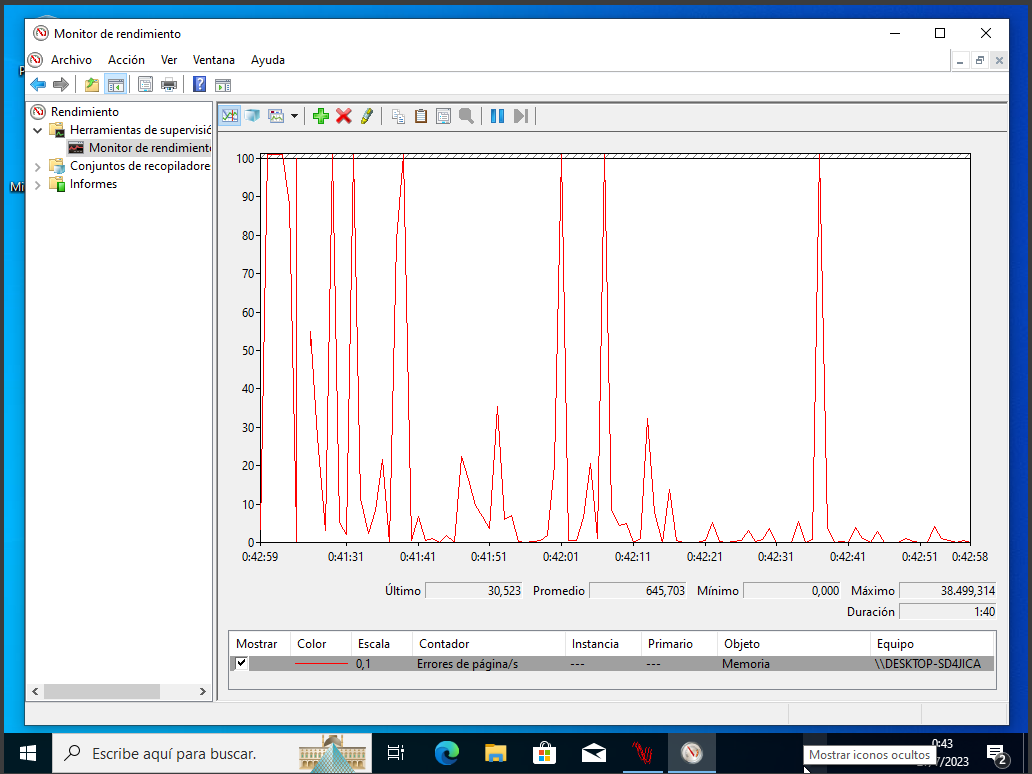
Logotipo

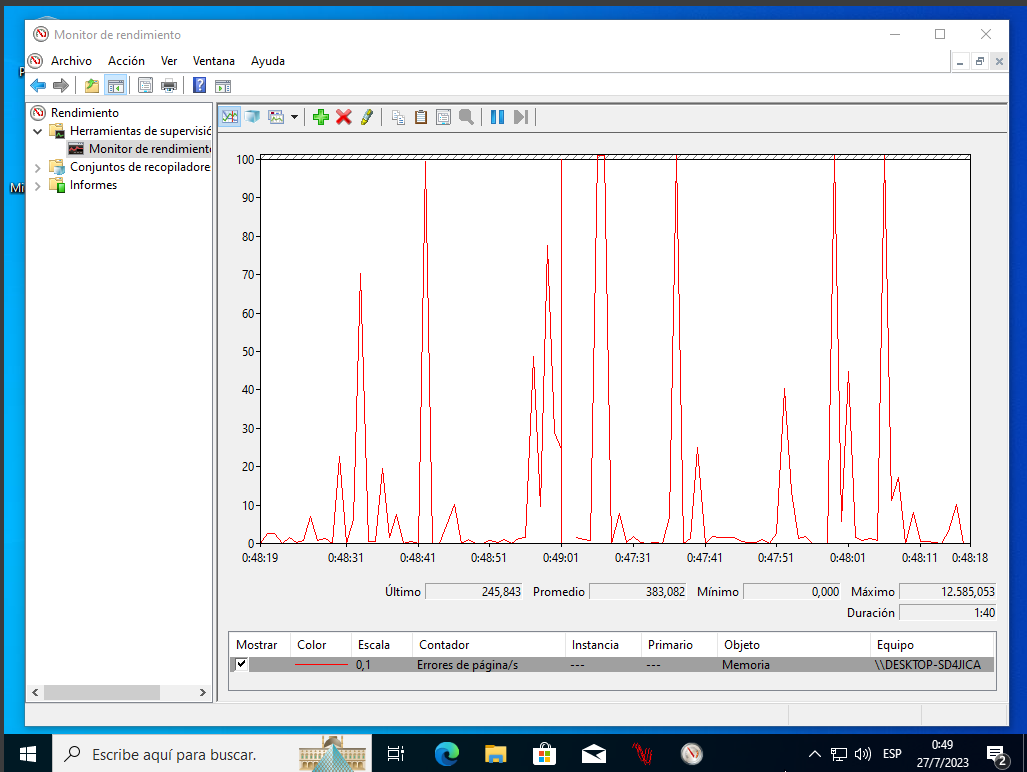
Descripción generada automáticamente

2 ram

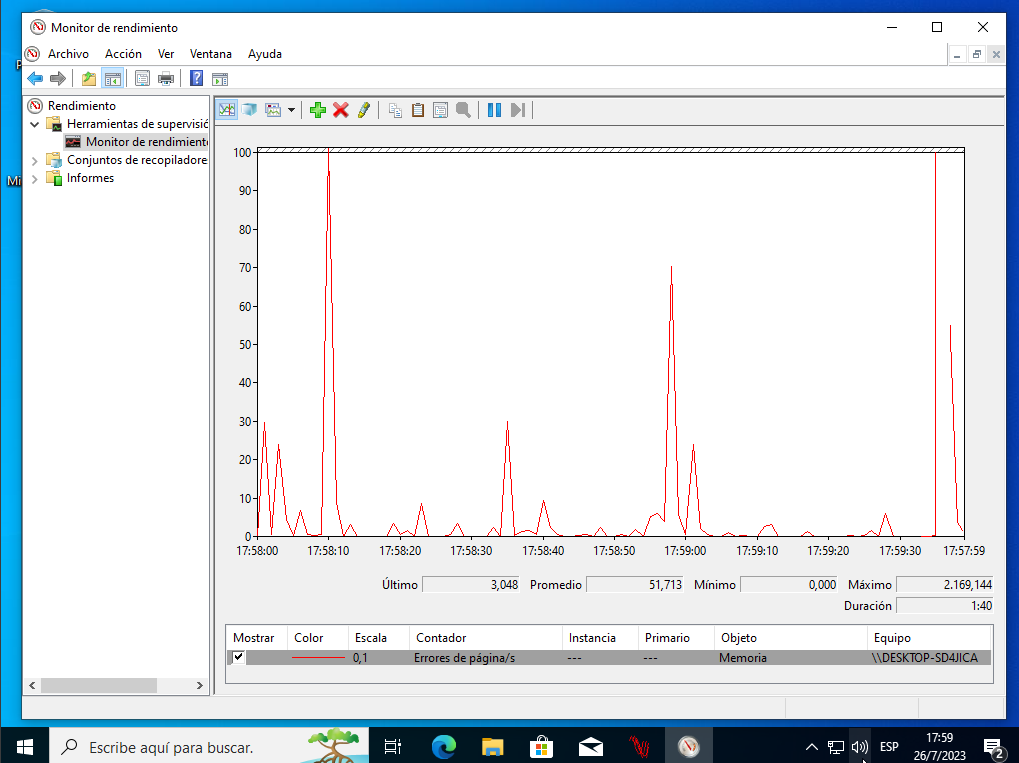
3RAM



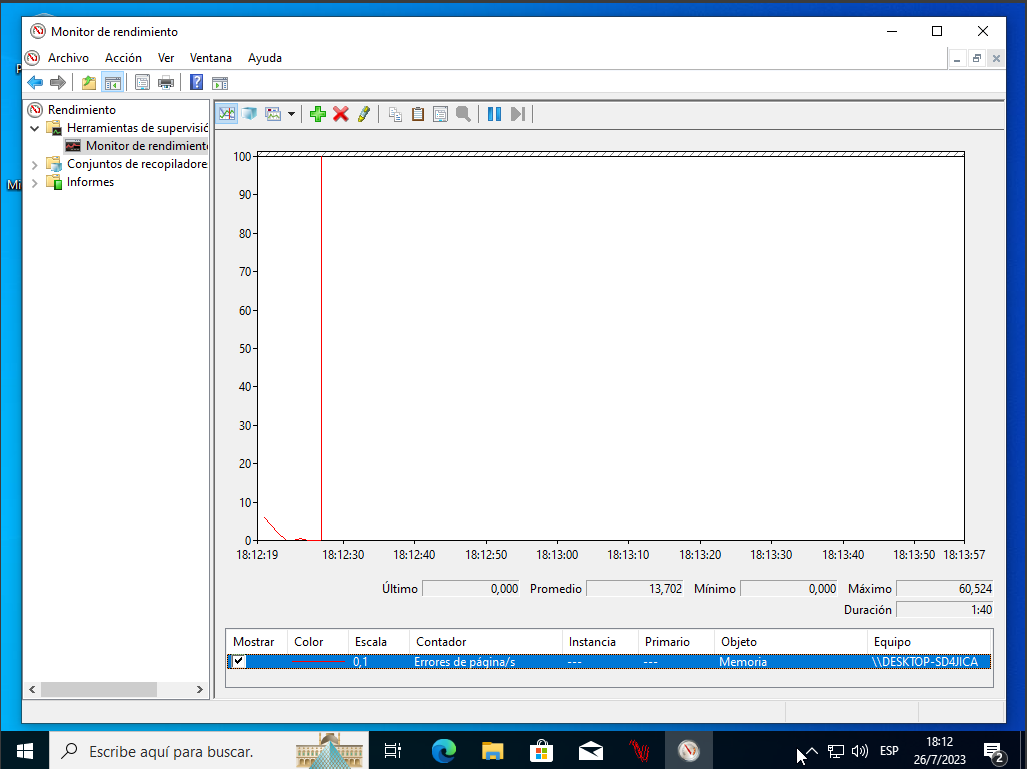
4RAM

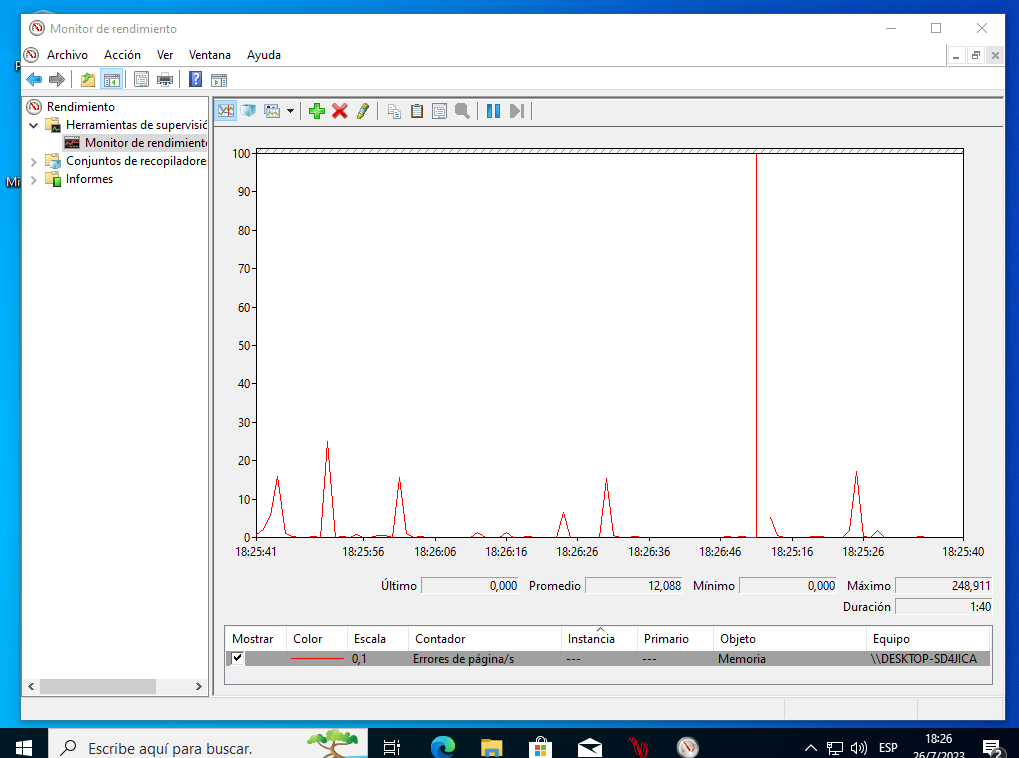


5 ram

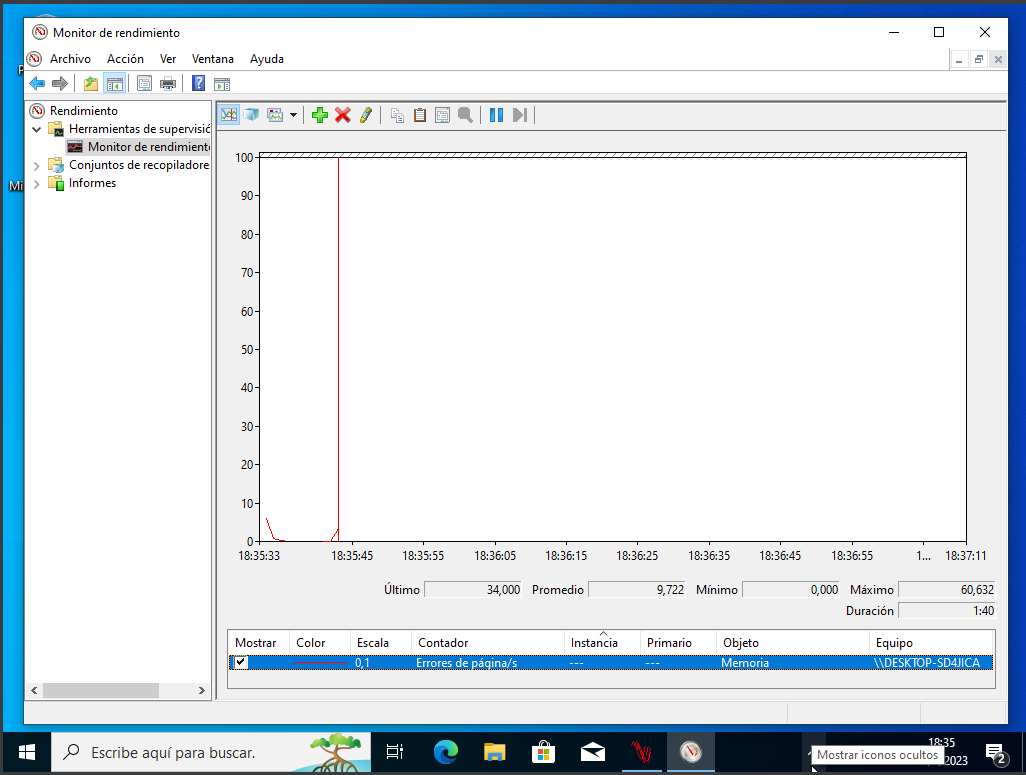


6ram

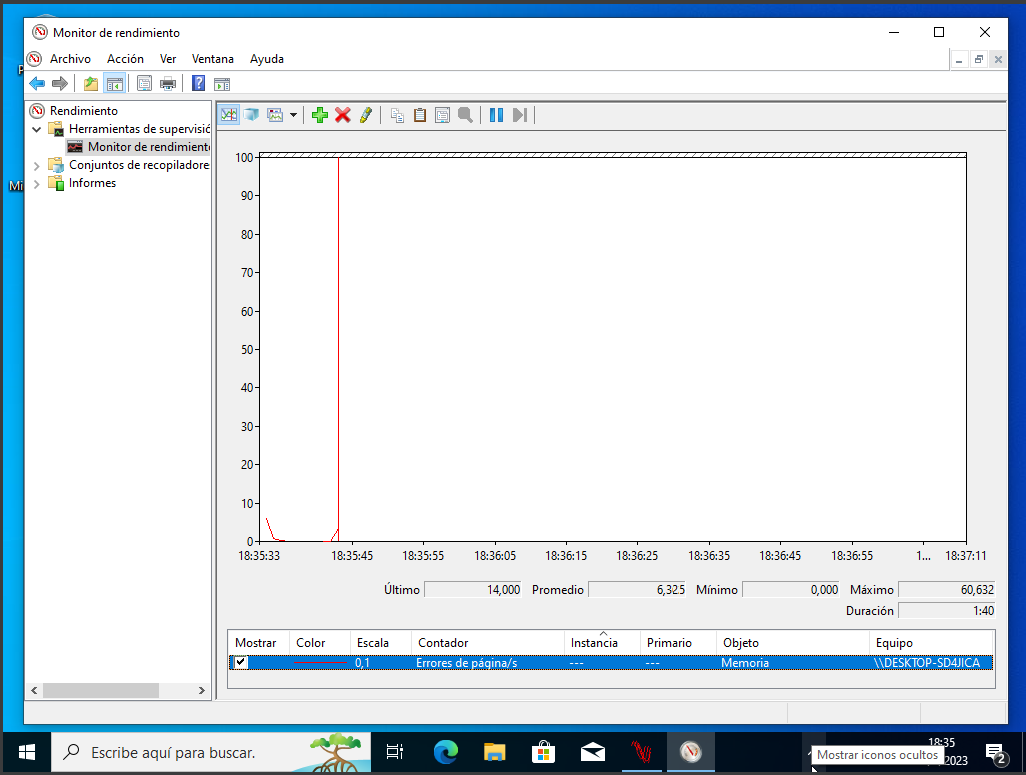


7 RAM  


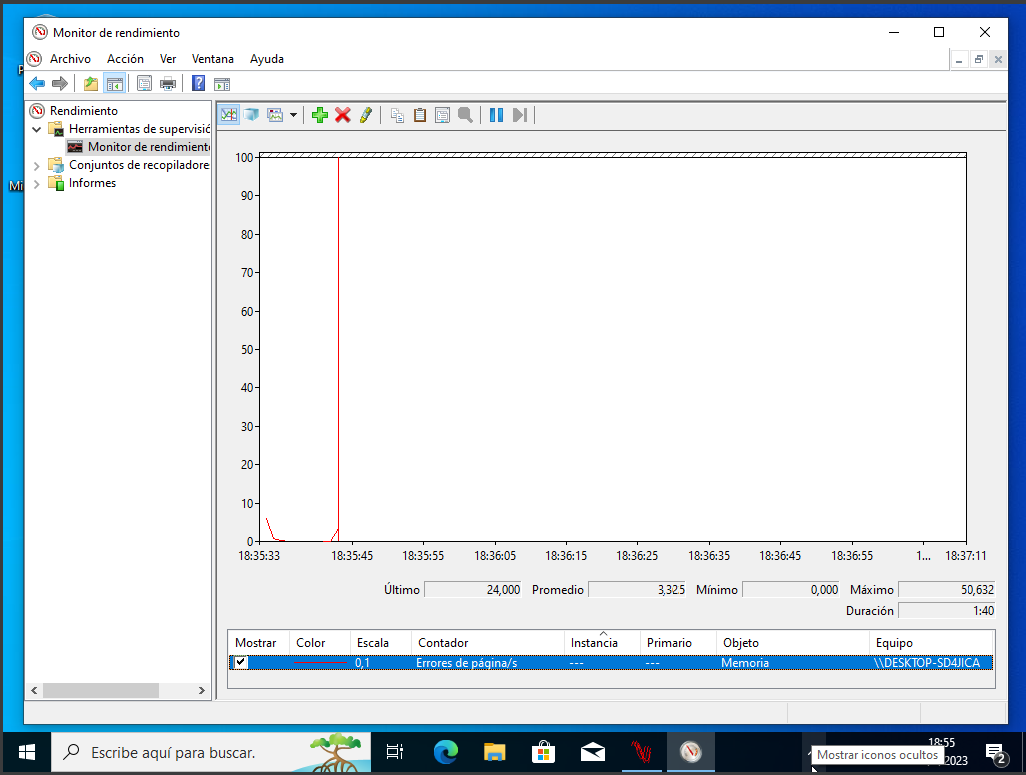
8 RAM



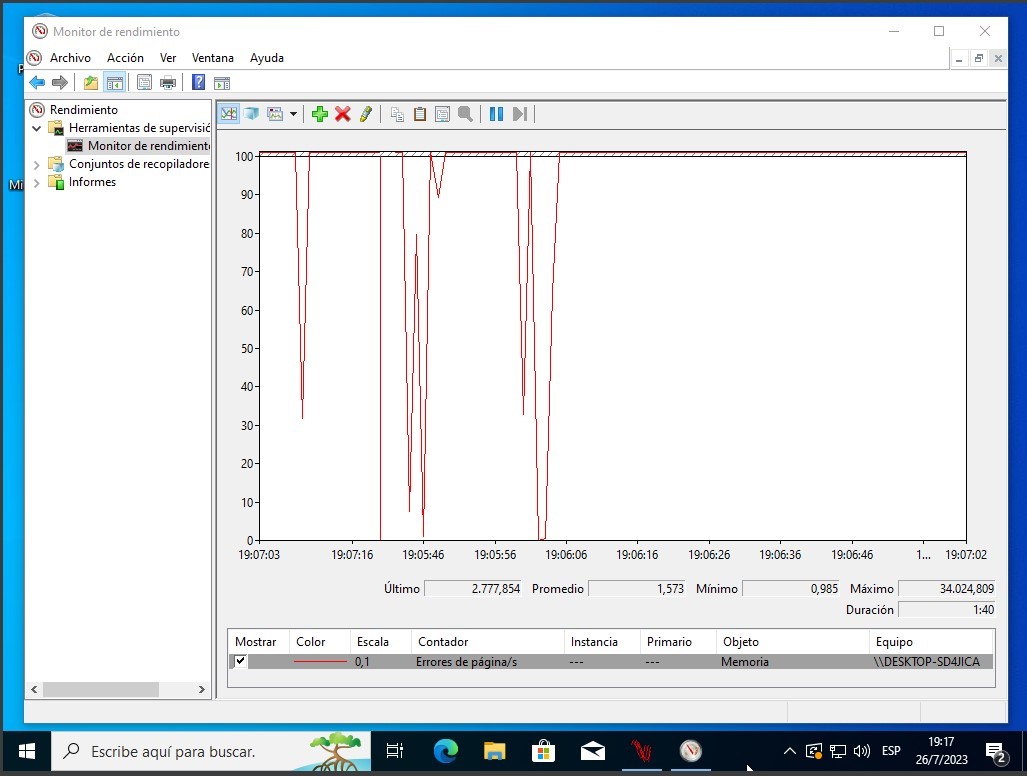
9Ram



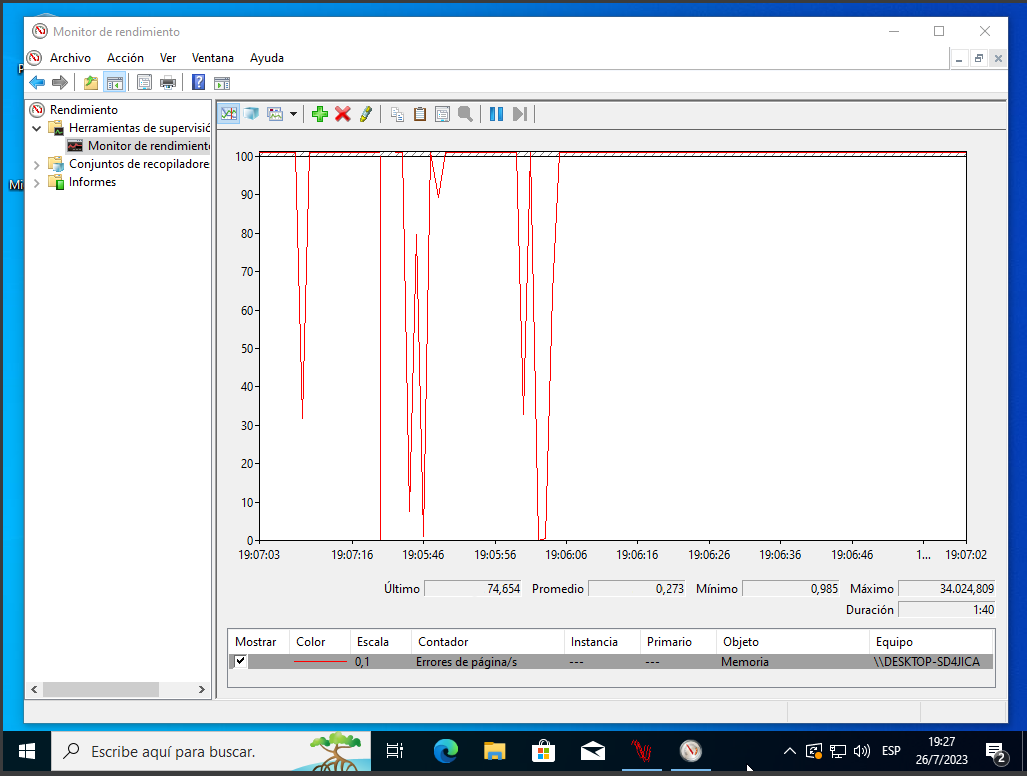
10RAM



11RAM



12RAM















































**ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**SOFTWARE**

**DEFINICIÓN DE TECNOLOGÍAS Y PILA DE PRODUCTO DEL PROYECTO**

**DOCENTE:**

ING. JULIO ROBERTO SANTILLAN CASTILLO

**ESTUDIANTE:**

JOSÉ LUIS GUAMÁN -7174

APLICACIONES INFORMÁTICAS II

8VO, SOFTWARE “A”

cursas

**MATERIA:**

**CURSO:**

19/11/2024

**Repositorio:** [**https://github.com/areascript22/Project-Indriver/tree/main/documentation/Product%20Backlog**](https://github.com/areascript22/Project-Indriver/tree/main/documentation/Product%20Backlog)

**Escala de prioridad.**

1. Muy alto.
2. Alto.
3. Medio.
4. Bajo.

**Historias de Usuario.**

|  |  |
| --- | --- |
| Historias de usuario #001: Registro e inicio de sesión de Pasajeros (RF-01) | |
| Como Pasajero, quiero registrarme en la aplicación usando mi correo electrónico y contraseña, para poder acceder y usar el servicio. | |
| Validación:   * El Pasajero puede iniciar sesión utilizando el número de celular. * El pasajero debe un código SMS en su celular. * El pasajero puede proporcionar el SMS para corroborar que se sea su numero de celular. * Si es la primera vez que el pasajero utiliza la app se le desplegara una pantalla para proporcionar sus datos como nombre e imagen de perfil. * Si el usuario inicia la app pero tiene datos registrado se le redirigirá directamente a la pagina de inicio. | Valor:  Prioridad: 1  Estimación: 7 h |

|  |  |
| --- | --- |
| Historias de usuario #002: Registro de Conductores (RF-02) | |
| Como administrador, quiero iniciar sesión usando mi correo electrónico y registrar conductores en la aplicación, para que el conductor pueda acceder y usar el servicio. | |
| Validación:   * El administrador puede iniciar sesión usando su correo y contraseña. * El administrador puede recuperar su contraseña en caso de olvidarla. * El administrador puede elegir la opción de registrar conductor en la app Administrado, la cual contendrá un formulario para ingresar todos los datos del conductor. * El conductor debe recibir un enlace en su correo electrónico para establecer su contraseña. * El enlace de establecimiento de contraseña esta disponible durante 6 horas. * La cuenta del conductor debe estar registrada en Firebase Authentication y los datos deben estar guardados en Firebase Firestore. | Valor:  Prioridad: 1  Estimación: 7 h |

|  |  |
| --- | --- |
| Historias de usuario #003: Inicio de Sesión (Conductor) (RF-02) | |
| Como Conductor registrado, quiero iniciar sesión con mi correo electrónico, para acceder a mi cuenta de forma segura. | |
| Validación:   * El Conductor puede iniciar sesión haciendo uso de su correo y contraseña previamente registradas por el administrador. * Las credenciales se validan antes de permitir el acceso. * El conductor puede recuperar o cambiar su contraseña si así lo desea utilizando su correo electrónico. * El conductor recibe feedback de cada interacción que haga. * Para recuperar la contraseña se le enviara un enlace al correo electrónico del Conductor. * El conductor puede seleccionar la opción “Cerrar sesión” si así lo desea. * El conductor interactúa mediante una interfaz amigable. | Valor:  Prioridad: 1  Estimación: 7 h |

|  |  |
| --- | --- |
| Historias de usuario #004: Gestión de Perfil (RF-04) | |
| Como pasajero, quiero poder ver y editar mi perfil para mantener actualizada mi información personal. | |
| Validación:   * El sistema proporciona una vista previa sobre los datos del pasajero. * El sistema permite navegar a la pantalla de edición de datos, en donde el pasajero podrá editar su imagen de perfil, nombre, apellido y número de celular. * Los cambios se reflejan en Firebase Firestore. * Cada campo esta validado para evitar datos vacíos. * Para la imagen de perfil, el sistema permite seleccionar una imagen de la galería o tomarse una foto. | Valor:  Prioridad: 2  Estimación: 5 h |

|  |  |
| --- | --- |
| Historias de usuario #005: Recuperación de Contraseña (RF-03) | |
| Como usuario, quiero recuperar mi contraseña en caso de que la olvide, para poder restablecerla y acceder a mi cuenta. | |
| Validación:   * El usuario recibe un código de verificación por correo o teléfono. *  Puede restablecer su contraseña usando el código. | Valor:  Prioridad: 2  Estimación: 6 h |

|  |  |
| --- | --- |
| Historias de usuario #006: Geolocalización en Tiempo Real (RF-05) | |
| Como cliente, quiero ver la ubicación en tiempo real de mi conductor durante el viaje para saber su posición actual. | |
| Validación:   * La ubicación se actualiza cada pocos segundos. * Se muestra un mapa interactivo con la posición de ambos (cliente y conductor). | Valor:  Prioridad: 1  Estimación: 14 h |

|  |  |
| --- | --- |
| Historias de usuario #007: Solicitud de Transporte (RF-06) | |
| Como cliente, quiero poder solicitar un viaje indicando mi punto de partida y destino para que un conductor me lleve. | |
| Validación:   * El usuario puede seleccionar la dirección de origen y destino mediante el mapa interactivo. * La solicitud se envía a los conductores cercanos. | Valor:  Prioridad: 1  Estimación: 15 h |

|  |  |
| --- | --- |
| Historias de usuario #008: Selección de Conductor (RF-07) | |
| Como cliente, quiero poder seleccionar un conductor de una lista de opciones para elegir el que prefiera. | |
| Validación:   * La lista de conductores se muestra con los perfiles de cada conductor. * El sistema asigna el viaje al conductor seleccionado. | Valor:  Prioridad: 2  Estimación: 8 h |

|  |  |
| --- | --- |
| Historias de usuario #009: Notificaciones en Tiempo Real (RF-08) | |
| Como cliente o conductor, quiero recibir notificaciones en tiempo real sobre el estado de los viajes para estar al tanto de los cambios. | |
| Validación:   * Las notificaciones se envían para cada cambio de estado (el conductor ha aceptado el viaje, el conductor ha llegado, el cliente está en camino). | Valor:  Prioridad: 1  Estimación: 11 h |

|  |  |
| --- | --- |
| Historias de usuario #010: Aceptación de Solicitud por el Conductor (RF-09) | |
| Como conductor, quiero poder aceptar las solicitudes de viaje que me interesen para realizar viajes. | |
| Validación:   * El conductor puede ver una lista de solicitudes activas con el perfil de cada cliente. * El sistema notifica al cliente cuando un conductor acepta su solicitud. | Valor:  Prioridad: 1  Estimación: 11 h |

|  |  |
| --- | --- |
| Historias de usuario #011: Calificación y Comentarios (RF-11) | |
| Como cliente o conductor, quiero poder dejar una calificación y comentario al finalizar un viaje para compartir mi experiencia. | |
| Validación:   * El usuario puede calificar de 1 a 5 estrellas. * Los comentarios se almacenan junto con la calificación. | Valor:  Prioridad: 3  Estimación: 7 h |

|  |  |
| --- | --- |
| Historias de usuario #012: Gestión de cuentas por el administrador (RF-15) | |
| Como administrador, quiero poder crear, actualiza, eliminar cuentas de conductores. | |
| Validación:   * El administrador puede navegar a la sección de gestionar conductores. * El administrador puede visualizar todos los conductores registrados. * El administrador puede editar la información de cualquier conductor. * El administrador puede eliminar un conductor específico. * Los cambios se ven reflejados en la base de datos. * Los datos actualizados se reflejan en la aplicación del conductor. | Valor:  Prioridad: 2  Estimación: 7 h |

|  |  |
| --- | --- |
| Historias de usuario #013: Generación de reportes (RF-16) | |
| Como administrador, quiero poder consultar un reporte detallado de los viajes completados y calificaciones recibidas, para evaluar el desempeño de los conductores. | |
| Validación:   * El administrador puede navegar a la sección de generar reportes. * El administrador puede descargar el reporte en formato pdf. | Valor:  Prioridad: 2  Estimación: 7 h |

**Historias técnicas.**

|  |  |
| --- | --- |
| Historia técnica #001: Integración con Firebase (RF-01, RF-02, RNF-02) | |
| Como desarrollador, necesito integrar Firebase Authentication, Firebase Firestore y Firebase Storage para la autenticación y registro del Cliente. | |
| Validación:   * El SDK de Firebase esta correctamente instalado Flutter con todos los paquetes requeridos tanto en las aplicaciones del Pasajero como en el Conductor.      * El SDK de Firebase esta correctamente instalado en Angular en la aplicación del Administrador. * En la plataforma de Firebase esta correctamente habilitados los servicios de Firestore, Realtime Database, Authentication. Además de la correcta creación de una cuenta en GCP (Google Cloud Platform) y la creación de una cuenta de facturación para poder usar servicio tanto de GCP como de Firebase. * Reglas de seguridad correctamente implementadas en todos los servicios habilitados en Firebase. * En la plataforma de Firebase se pueden ver 5 aplicaciones integradas: Conductor (Android y IOS), Pasajero (Android y IOS), Administrador (Web). | Valor:  Prioridad: 1  Estimación: 14 h |

|  |  |
| --- | --- |
| Historia técnica #002: Actualización de Ubicación en Tiempo Real con Firebase Realtime Database (RF-05, RNF-01) | |
| Como desarrollador, quiero implementar la actualización de la ubicación en tiempo real usando Firebase Realtime Database para proporcionar un seguimiento preciso. | |
| Validación:   * La posición tanto de cliente y conductor se actualiza en el mapa. * La actualización es visible en un mapa interactivo. | Valor:  Prioridad: 1  Estimación: 17 h |

|  |  |
| --- | --- |
| Historia técnica #003: Compatibilidad del Sistema Operativo (RNF-03, RNF-04) | |
| Como desarrollador, quiero optimizar la aplicación para ser compatible con múltiples versiones de iOS y Android. | |
| Validación:   * La aplicación funciona correctamente en dispositivos con diferentes versiones de SO. | Valor:  Prioridad: 2  Estimación: 12 h |

|  |  |
| --- | --- |
| Historia técnica #004: Diseño de la base de datos (OR-01) | |
| Como desarrollador, quiero diseñar la estructura de la base de datos para los Pasajeros, Conductores y Administrador, además de la estructura json para la comunicación en tiempo real. | |
| Validación:   * No deben existir campos duplicados. * Almacene los datos de los usuarios en una única colección (usuarios, pasajeros, admin) para lograr escalabilidad y simplicidad. * Implementar createdAt y updatedAt para rastrear cambios y ordenar registros. * Todas las claves deben estar entre comillas dobles (""). * Los pares clave-valor deben separarse con comas. * No se permiten comas finales. * ¿Los conductores pueden ver las solicitudes pendientes? * ¿Los pasajeros pueden ver a los conductores que aceptaron sus solicitudes? * ¿Hay alguna manera de hacer un seguimiento del estado de las solicitudes (por ejemplo, pendientes, aceptadas)? | Valor:  Prioridad: 1  Estimación: 7 h |

|  |  |
| --- | --- |
| Historia técnica #005: Refactorización del código | |
| Como desarrollador, quiero refactorizar el código una vez culminada con el proyecto para de esta manera garantizar un código fuente legible, mantenible, escalable y de calidad. | |
| Validación:   * Eliminación de código no utilizado o redundante. * El nombre de los archivos debe seguir la convención snake\_case. El nombre de las clases debe seguir la convención PascalCase. El nombre de las variables debe seguir la convención camelCase. | Valor:  Prioridad: 2  Estimación: 7 h |

|  |  |
| --- | --- |
| Historia técnica #006: Aprender mediante tutoriales | |
| Como desarrollador, quiero dedicar tiempo a aprender tecnologías y frameworks específicos (Google map API, Firebase, Pasarelas de pago) a través de tutoriales, de modo que pueda implementar todas las funcionalidades de manera efectiva. | |
| Validación:   * Cursos identificados en plataformas como YouTube, Udemy, etc. En los siguientes temas:   + Google Maps API.   + Google Directions API.   + Google Geolocation API.   + Firebase Authentication.   + Firebase Firestore.   + Firebase Storage   + Firebase Realtime Database.   + Pasarelas de pago.   + Flutter.   + Angular. * El conocimiento adquirido debe ser aplicado en la implementación de cada funcionalidad del sistema. | Valor:  Prioridad: 1  Estimación: 14 h |

**Product backlog.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Identificador de la historia.** | **Historia de usuario/ técnica** | **Horas** | **Sprint** | **Estado** | **Prioridad** | **Comentarios** |
| Historia técnica #001 | Integración con Firebase (RF-01, RF-02, RNF-02) | 14 h | 1 | Completado | **1** |  |
| Historia técnica #004 | Diseño de la base de datos (OR-01) | 7 h | 2 | Completado | **1** |  |
| Historias de usuario #002 | Registro de Conductores (RF-02) | 7 h | 2 | Completado | **1** |  |
| Historias de usuario #003 | Inicio de Sesión (Conductor) (RF-02) | 14 h | 3 | Completado | **1** |  |
| Historias de usuario #001 | Registro e inicio de sesión de Pasajeros (RF-01) | 5 h | 4 | Pendiente | **1** |  |
| Historias de usuario #004 | Gestión de Perfil (RF-04) | 4 h | 4 | Pendiente | **2** |  |
| Historias de usuario #012 | Gestión de cuentas por el administrador (RF-15) | 5h | 12 | Pendiente | **2** |  |
| Historia técnica #002 | Actualización de Ubicación en Tiempo Real con Firebase Realtime Database (RF-05, RNF-01) | 14 h | 5 | Pendiente | **1** |  |
| Historias de usuario #005 | Geolocalización en Tiempo Real (RF-05) | 14 h | 6 | Pendiente | **2** |  |
| Historias de usuario #006 | Solicitud de Transporte (RF-06) | 14 h | 7 | Pendiente | **1** |  |
| Historias de usuario #007 | Selección de Conductor (RF-07) | 8 h | 8 | Pendiente | **1** |  |
| Historias de usuario #008 | Notificaciones en Tiempo Real (RF-08) | 6 h | 8 | Pendiente | **2** |  |
| Historias de usuario #009 | Aceptación de Solicitud por el Conductor (RF-09) | 14 h | 9 | Pendiente | **1** |  |
| Historias de usuario #010 | Calificación y Comentarios (RF-11) | 7 h | 10 | Pendiente | **1** |  |
| Historias de usuario #002 | Registro de Conductores (RF-02) | 7 h | 10 | Pendiente | **1** |  |
| Historia técnica #003 | Compatibilidad del Ssitema Operativo (RNF-03, RNF-04) | 14 h | 11 | Pendiente | **2** |  |
|  |  |  |  |  |  |  |
| Historias de usuario #013 | Generación de reportes (RF-16) | 7h | 12 | Pendiente | **2** |  |

**Comparación de tecnologías**

**Lenguaje de Programación**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lenguaje** | **Dart** | **JavaScript** | **Kotlin** | **Elección Final** |
| **Pros** | Optimizado para Flutter, rápido, fácil de depurar, soporte sólido para aplicaciones multiplataforma. | Amplia comunidad, versátil, soporte extensivo para bibliotecas y frameworks. | Específico para Android, gran rendimiento y herramientas avanzadas para Android. | **Dart** |
| **Contras** | Comunidad más pequeña, enfoque limitado fuera de Flutter. | Puede ser complicado para aplicaciones móviles, gestión de asincronía compleja. | Solo aplicable para Android, no es multiplataforma. | Comunidad en crecimiento. |
| **Uso Ideal** | Aplicaciones móviles multiplataforma con alto rendimiento. | Aplicaciones web, híbridas, o con interacción ligera. | Apps Android nativas. | Apps modernas multiplataforma. |

**IDE/Editor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IDE/Editor** | **Visual Studio Code** | **Android Studio** | **Xcode** | **Elección Final** |
| **Pros** | Ligero, extensible, multiplataforma, ideal para desarrollo rápido. | Herramientas avanzadas para pruebas y depuración de Android. | Optimizado para desarrollo de apps iOS. | **Visual Studio Code** |
| **Contras** | Depuración limitada en simuladores complejos. | Requiere hardware potente, puede ser pesado. | Exclusivo para macOS, curva de aprendizaje inicial para usuarios no iOS. | Algunas funciones requieren extensiones adicionales. |
| **Uso Ideal** | Desarrollo ágil y multiplataforma. | Desarrollo de aplicaciones Android nativas. | Apps nativas para iOS. | Desarrollo de apps multiplataforma con Flutter. |

**Framework para la App Móvil**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Framework** | **Flutter** | **React Native** | **Ionic** | **Elección Final** |
| **Pros** | UI nativa, único código base, excelente rendimiento. | Amplia comunidad, buena compatibilidad con JS, compatible con librerías de terceros. | Compatible con web y móvil, fácil de implementar. | **Flutter** |
| **Contras** | Tamaño inicial grande, curva de aprendizaje inicial. | Depende de librerías externas para optimización de rendimiento. | Rendimiento inferior al nativo. | Curva de aprendizaje inicial. |
| **Uso Ideal** | Apps móviles de alto rendimiento con experiencia nativa. | Apps híbridas con funcionalidad estándar. | Apps ligeras, proyectos pequeños. | Apps móviles profesionales que requieran buen diseño y rendimiento. |

**Framework para la App Web**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Framework** | **React** | **Angular** | **Vue.js** | **Elección Final** |
| **Pros** | Popularidad global, fácil integración con bibliotecas de terceros, componentes reutilizables. | Escalabilidad, buenas prácticas integradas, soporte sólido de Google. | Ligero, fácil de aprender, curva de aprendizaje amigable. | **Angular** |
| **Contras** | Requiere configuración extra para backend. | Curva de aprendizaje empinada, requiere mayor conocimiento inicial. | Ecosistema más pequeño comparado con React y Angular. | Mayor tiempo para dominar características avanzadas. |
| **Uso Ideal** | Proyectos pequeños y medianos con componentes dinámicos. | Aplicaciones grandes y complejas con altos requisitos de escalabilidad. | Proyectos ligeros o con plazos de desarrollo cortos. | Apps escalables con gran cantidad de usuarios. |

**Base de Datos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Base de Datos** | **Firebase (Firestore)** | **Cassandra** | **MongoDB** | **Elección Final** |
| **Pros** | Integración nativa con Flutter, escalabilidad en tiempo real, solución sin servidor. | Diseñada para ofrecer disponibilidad continua sin puntos únicos de falla. Esto la hace ideal para sistemas que necesitan operar 24/7. | Escalable, flexible, fácil de usar para datos no estructurados. | **Firebase (Firestore)** |
| **Contras** | Limitado para consultas relacionales complejas. | Requiere experiencia técnica para configurar y mantener correctamente. Las configuraciones incorrectas pueden afectar el rendimiento y la disponibilidad. | No ideal para datos altamente estructurados. | Limitaciones en migración de datos. |
| **Uso Ideal** | Aplicaciones con datos en tiempo real y necesidades de sincronización instantánea. | Servicios de registro de actividades, aplicaciones IoT, y sistemas de monitoreo que generan datos continuamente. | Aplicaciones con alto volumen de datos y necesidades de escalabilidad horizontal. | Aplicaciones móviles modernas y multiplataforma. |

**Razones para la elección final**

**Dart:** Diseñado específicamente para Flutter, lo que asegura la mejor compatibilidad y rendimiento para aplicaciones móviles multiplataforma.

**Flutter:** Proporciona un desarrollo rápido con un diseño visual atractivo y un excelente rendimiento nativo.

**Angular:** Es especialmente útil para proyectos empresariales o aplicaciones que requieren una arquitectura robusta, una gran cantidad de funcionalidades integradas y un enfoque modular y limpio.

**Firebase:** Ofrece sincronización en tiempo real, escalabilidad, y una fácil integración con Flutter, perfecto para aplicaciones de transporte como la tuya.

**Visual Studio Code:** Ligero, multiplataforma, y extensible, ideal para desarrolladores que trabajan con Flutter.