**SCRUM: Sprint Backlog**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprints.** | **Días** | **Actividades** | **Estado** |
| 1 | Martes | Capacitación sobre Firebase Authentication y Firebase Firestore | Completado |
| Miércoles | Capacitación sobre Firebase Realtime Database y Reglas de Seguridad. | Completado |
| Jueves | Capacitación sobre la integración de los servicios de Firebase tanto en el Framawork Flutter como en Angular. Además del uso básico de GCP (Google Cloud Platform). | Completado |
| Viernes | Integración del SDK de Firebase en la app Flutter tanto del Cliente como del Conductor. | Completado |
| Sábado | Integración del SDK de Firebase en la app del Administrador en Angular. | Completado |
| Domingo | Creación de la cuenta de facturación en Google Cloud Platform para actualización del proyecto Firebase al Plan Blaze, de este modo poder utilizar Firebase Storage. | Completado |
| Lunes | Implementación de las reglas de seguridad en Firebase Firestore, Realtime Database y Firebase Storage. | Completado |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprints.** | **Días** | **Actividades** | **Estado** |
| 2 | Martes | Diseño de la base de datos tanto para el Conductor, Pasajero y Administrador. | Incompleto |
| Miércoles |
| Jueves | Diseño de la estructura json para la comunicación en tiempo real entre Conductor y Pasajero. | Completado |
| Viernes | Implementación de la función de Inicio de sesión como administrador en Angular. | Completado |
| Sábado | Implementación de la función “Recuperar Contraseña” para el Administrador. | Completado |
| Domingo | Diseño de la interfaz y estructura básica de la aplicación separando responsabilidades para gestionar Clientes y Conductore. | Completado |
| Lunes | Implementación del registro de Conductores mediante un formulario en Angular. | Completado |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprints.** | **Días** | **Actividades** | **Estado** |
| 3 | Martes | Capacitación en la arquitectura MVVM (Model-View-ViewModel). | Completado |
| Miércoles | Capacitación en el principio Feature-Based Development para la correcta estructuración del proyecto. | Completado |
| Jueves | Resolución del problema de compatibilidad de los paquetes de Firebase con la versión de Kotlin y el SDK de Flutter. | Completado |
| Viernes | Definición de la paleta de colores y el archivo d configuración del tema para la aplicación. | Completado |
| Sábado | Implementación del inicio de sesión del conductor. | Completado |
| Domingo | Implementación de la funcionalidad “Recuperación de contraseña” | Completado |
| Lunes | Implementación del “Drawer” para la implementación del “Cierre de sesión” | Completado |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprints.** | **Días** | **Actividades** | **Estado** |
| 4 | Martes | Implementación de Inicio de Sesión mediante número de celular. | Completado |
| Miércoles |
| Jueves | Implementación de la funcionalidad “Cerrar sesión”. | Completado |
| Viernes | Implementación de la página “Editar información” del conductor. | Completado |
| Sábado | Implementación de la función Listar Conductores. | Completado |
| Domingo | Implementación de la funcionalidad “Eliminar conductores” | Completado |
| Lunes | Implementación de la funcionalidad “Editar conductores” | Completado |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprints.** | **Días** | **Actividades** | **Estado** |
| 5 | Martes | Documentación de la api Geocoding de la plataforma de Google Cloud Console. | Completado |
| Miércoles | Documentación de integracion del mapa de google en Flutter. | Completado |
| Jueves | Implementación de la funcionalidad de seleccionar un punto en el mapa y convertir las coordenadas seleccionadas en una dirección legible por el usuario mediante la api Geocoding. | Completado |
| Viernes | Implementación de Marcadores mostrados en cada punto seleccionado. | Completado |
| Sábado | Implementación de la ruta dibujada entre los puntos de recogida y de destino. | Completado |
| Domingo | Implementación del Bottom Sheet para mostrar datos de los puntos seleccionados, así como el tiempo estimado del viaje basado en la ruta. | Completado |
| Lunes | Implementación del Botón “Solicitar un conductor” | Completado |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprints.** | **Días** | **Actividades** | **Estado** |
| 6 | Martes | Conductor: Implementación de la funcionalidad en que cada que la app se ejecuta, sus coordenadas físicas se actualizan automáticamente en Firebase Realtime Database. | Completado |
| Miércoles | Conductor: Implementación de la cola, un conductor puede reservar un puesto en la cola de 6 vehículos, así como liberar su puesto para que automáticamente el anterior pase a ocupar su puesto. | Completado |
| Jueves |
| Viernes | Pasajero: Cuando el sistema se le asigna un conductor, se muestra su información en la app. | Completado |
| Sábado |
| Domingo | Funcionalidad tanto en la app de conductor como de pasajero para poder comunicarse mediante Whatsapp entre ellos. | Completado |
| Lunes |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprints.** | **Días** | **Actividades** | **Estado** |
| 7 | Martes | Conductor: Funcionalidad, cada que la app se ejecute se inicializa un listener de ubicación física basada en la señal GPS del celular. | Completado |
| Miércoles | Cada que el GPS detecte un cambio de posición se va a actualizar automáticamente en Firebase Reltime Database. | Completado |
| Jueves | Funcionalidad de dar la posibilidad de intercambiar entre disponible y no disponible. Si el conductor esta disponible será tomado en cuenta para la solicitudes de transporte. | Completado |
| Viernes | Pasajero: Cuando solicita un vehículo el sistema retornará el primer conductor en la cola basándose en el campo Timestamp. | Completado |
| Sábado | Una vez determinado el conductor para el pasajero se escribe los datos del pasajero como un nodo hijo del conductor en Firebase Realtime Database. | Completado |
| Domingo | Implementación de la funcionalidad, si es que a app del conductor se desconecta por algún motivo de la base de datos se eliminan automáticamente sus datos de Firebase Realtime Database para evitar datos innecesarios. | Completado |
| Lunes |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprints.** | **Días** | **Actividades** | **Estado** |
| 8 | Martes | Una vez determinado el conductor para una petición de transporte, automáticamente se le muestra al conductor un mapa con la ruta y los puntos tanto de recogida como de destino así como la información del pasajero como nombre, foto de perfil, calificación en estrellas, etc. | Completado |
| Miércoles | Cuando el conductor haya llegado al punto de recogida puede enviar una notificación en tiempo real al pasajero para hacerle saber que esta esperando en el punto. | Completado |
| Jueves |
| Viernes | De la misma manera el pasajero puede notificar al conductor que esta llegando hacia el vehículo para poder iniciar la ruta hacia el punto de destino. | Completado |
| Sábado |
| Domingo | El conductor puede finalizar el viaje presionando un botón, y el pasajero es notificado que el viaje a concluido. | Completado |
| Lunes |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprints.** | **Días** | **Actividades** | **Estado** |
| 9 | Martes | Diseñar la estructura de Firestore para almacenar las calificaciones y comentarios. | Completado |
| Miercoles | Crear un RatingDialog (Ventana modal) en Flutter con estrellas y campo de texto para comentarios. | Completado |
| Jueves | Implementar lógica para guardar la calificación en Firestore al presionar "Enviar". | Completado |
| Viernes | Validar que un usuario solo pueda calificar una vez por viaje. | Completado |
| Sabado | Calcular y mostrar el promedio de calificaciones en el perfil de un usuario. | Completado |
| Domingo | Desplegar los cambios en Firebase y realizar pruebas. | Completado |
| Lunes |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprints.** | **Días** | **Actividades** | **Estado** |
| 10 | Martes | Revisar y definir la versión mínima y target de Android en android/app/build.gradle. | Completado |
| Miercoles | Actualizar todas las dependencias de Flutter a sus versiones más recientes compatibles con la versión mínima de Android. | Completado |
| Jueves | Identificar y corregir posibles incompatibilidades de código con versiones antiguas de Android. | Completado |
| Viernes | Probar la app en emuladores y dispositivos físicos con diferentes versiones de Android. | Completado |
| Sabado | Resolver errores de compatibilidad con permisos y configuraciones en AndroidManifest.xml. | Completado |
| Domingo | Verificar el correcto funcionamiento de los servicios de ubicación en diferentes versiones de Android. | Completado |
| Lunes |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprints.** | **Días** | **Actividades** | **Estado** |
| 11 | Martes | Diseñar la estructura de Firestore para almacenar el historial de viajes. | Completado |
| Miercoles | Implementar la lógica para guardar los datos de cada viaje al finalizar. | Completado |
| Jueves | Crear una pantalla en Flutter para mostrar el historial de viajes con lista y detalles. | Completado |
| Viernes | Implementar la consulta de historial en Firestore para obtener los viajes de cada usuario. | Completado |
| Sabado | Optimizar la carga de datos para mejorar rendimiento en listas largas. | Completado |
| Domingo | Realizar pruebas tanto en la app de conductor como del pasajero. | Completado |
| Lunes |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprints.** | **Días** | **Actividades** | **Estado** |
| 12 | Martes | Identificar código redundante o innecesario en el proyecto. | Completado |
| Miercoles | Modularizar widgets y funciones reutilizables. | Completado |
| Jueves | Revisar dependencias y optimizar paquetes utilizados. | Completado |
| Viernes | Configurar Google Play Console y crear el proyecto para la app. | Completado |
| Sabado | Generar y firmar el archivo .aab correctamente. | Completado |
| Domingo | Subir la app a Google Play en el canal de pruebas internas. | Completado |
| Lunes | Agregar testers y distribuir la app para pruebas. | Completado |