



**INTEGRANTES:**

**Carlos Guzmán Oyarzun**

**Daniel González Chávez**

**Rodrigo Berrios Castillo**

**PROFESOR:**

**Fabian Alejandro Alcántara Guajardo**

Contenido

[Información del proyecto 3](#_heading=h.1y810tw)

[Patrocinador / Patrocinadores 3](#_heading=h.4i7ojhp)

[Propósito y justificación del proyecto 4](#_heading=h.2xcytpi)

[Descripción del proyecto y entregables 4](#_heading=h.1ci93xb)

[Requerimientos de alto nivel 6](#_heading=h.3whwml4)

[Requerimientos del producto 7](#_heading=h.2bn6wsx)

[Requerimientos del proyecto 8](#_heading=h.qsh70q)

[Objetivos 9](#_heading=h.3as4poj)

[Premisas y restricciones 9](#_heading=h.1pxezwc)

[Premisas 9](#_heading=h.jndurycofnow)

[Restricciones 10](#_heading=h.n0nxf82w3sz4)

[Riesgos iniciales de alto nivel 10](#_heading=h.49x2ik5)

[Riesgos Iniciales de Alto Nivel 11](#_heading=h.8hu47d3gy81p)

[Cronograma de hitos principales 12](#_heading=h.2p2csry)

[Presupuesto estimado 13](#_heading=h.147n2zr)

[Indicadores de Éxito 13](#_heading=h.3o7alnk)

[Lista de Interesados (stakeholders) 17](#_heading=h.23ckvvd)

[Requisitos de aprobación del proyecto 18](#_heading=h.ihv636)

[Asignación del gerente de proyecto y nivel de autoridad 19](#_heading=h.32hioqz)

[Gerente de proyecto 19](#_heading=h.1hmsyys)

[Niveles de autoridad 20](#_heading=h.41mghml)

[Personal y recursos preasignados **20**](#_heading=h.2grqrue)

# **Información del proyecto**

**Datos**

|  |  |
| --- | --- |
| Empresa / Organización | FlashMatch |
| Proyecto | FlashMatch - Plataforma digital para la organización y gestión de partidos de fútbol, reserva de canchas y gestión de equipos deportivos. |
| Fecha de preparación | 08 de octubre de 2024 |
| Cliente | Usuarios de la plataforma (futbolistas amateur, administradores de canchas deportivas) |
| Patrocinador principal | Carlos Guzmán Oyarzún  Cargo: Jefe de Proyecto, Responsable del desarrollo del sistema |
| Gerente de proyecto | Fabian alcantara  Escuela de Informática y Telecomunicaciones  Duoc UC |

## **Patrocinador / Patrocinadores**

| **Nombre** | **Cargo** | **Departamento / División** | **Rama ejecutiva (Vicepresidencia)** |
| --- | --- | --- | --- |
| Fabián Alcántara | Docente | Coordinación de Capstone | Escuela de Informática y Telecomunicaciones, Duoc UC |

# **Propósito y justificación del proyecto**

|  |
| --- |
| **Propósito del Proyecto:**  El propósito principal del proyecto FlashMatch es desarrollar una plataforma digital integral que facilite la organización de partidos de fútbol, la reserva de canchas deportivas y la gestión de equipos. Esta solución tiene como objetivo mejorar la experiencia de los usuarios al ofrecerles una herramienta eficiente y sencilla para gestionar sus actividades deportivas, optimizando la reserva de espacios y la coordinación de eventos. FlashMatch está diseñado para atender tanto a los jugadores amateur como a los administradores de canchas deportivas, creando un ecosistema que conecta ambos lados de la demanda deportiva.  **Justificación del Proyecto:**  En Chile, el fútbol es una de las actividades deportivas más populares, y la demanda de canchas para la práctica del deporte sigue en aumento. Sin embargo, la gestión de reservas y la organización de partidos sigue siendo un proceso manual y fragmentado, generando frustraciones entre los jugadores y los administradores. Según estudios recientes, un 45.8% de los partidos organizados se cancelan por problemas logísticos, como la falta de jugadores o la dificultad para coordinar espacios disponibles. Esta situación revela una clara oportunidad de optimización mediante el uso de una plataforma tecnológica que centralice estos procesos.  FlashMatch ofrece una solución integral a esta problemática, proporcionando a los usuarios la capacidad de gestionar todas sus actividades deportivas desde una sola aplicación móvil. Los administradores de canchas podrán optimizar la gestión de sus instalaciones, incrementando la ocupación y reduciendo los errores de reserva. Además, los jugadores podrán crear equipos, invitar a otros participantes y organizar partidos de manera más eficiente, mejorando la coordinación y la experiencia deportiva.  A largo plazo, FlashMatch busca consolidarse como la plataforma líder para la gestión de eventos deportivos, con posibilidades de expansión a otros deportes y regiones. Esto no solo beneficiará a los usuarios, sino que también generará nuevas oportunidades de negocio para los administradores de canchas deportivas y patrocinadores interesados en el deporte amateur. |

# 

# **Descripción del proyecto y entregables**

|  |
| --- |
| **Descripción del Proyecto:**  FlashMatch es una plataforma digital integral diseñada para optimizar la gestión de partidos de fútbol, facilitar la reserva de canchas deportivas, y simplificar la gestión de equipos. El proyecto incluye tanto una aplicación móvil como una página web de presentación, que permitirá a los usuarios acceder fácilmente a las funcionalidades principales de la plataforma.  La aplicación móvil está orientada a jugadores de fútbol amateur, permitiéndoles organizar partidos, gestionar equipos, visualizar estadísticas deportivas y reservar canchas de manera eficiente. Por otro lado, la página web sirve como un punto de entrada comercial, permitiendo a nuevos usuarios conocer la plataforma y descargar la app móvil.  El desarrollo del proyecto sigue la metodología ágil Scrum, organizada en sprints que permiten entregar funcionalidades incrementales y mejoras iterativas. El objetivo es ofrecer una solución de alta calidad que pueda adaptarse a las necesidades del mercado deportivo, con una arquitectura escalable que permite agregar nuevas funcionalidades, como la incorporación de más deportes y la gestión de torneos en futuras fases.  **Entregables del Proyecto:**   1. **Aplicación Móvil (Ionic):**    * **Funcionalidad de registro e inicio de sesión:** Los usuarios pueden crear cuentas o iniciar sesión a través de la autenticación tradicional o mediante Google.    * **Sistema de reserva de canchas:** Los usuarios pueden buscar canchas disponibles, ver horarios en tiempo real y hacer reservas.    * **Gestión de equipos y jugadores:** Los usuarios pueden crear y gestionar equipos, invitar a jugadores y ver estadísticas de su rendimiento.    * **Notificaciones en tiempo real:** Recordatorios de partidos, confirmación de reservas, y cambios en los horarios o equipos.    * **Panel de estadísticas de rendimiento:** Visualización de métricas deportivas personalizadas como velocidad, resistencia y desempeño táctico. 2. **Página Web de Presentación (Angular):**    * **Página de aterrizaje comercial:** Introduce la plataforma a posibles usuarios y socios comerciales, permitiendo la descarga de la aplicación móvil.    * **Información sobre las funcionalidades de la plataforma:** Explica las principales características, beneficios y el impacto que ofrece FlashMatch.    * **Contacto y soporte técnico:** Proporciona canales de contacto para asistencia técnica o consultas sobre el uso de la plataforma. 3. **Back-End (NestJS):**    * **APIs para la gestión de reservas y equipos:** APIs que gestionan la interacción entre la aplicación móvil y la base de datos.    * **Base de datos (PostgreSQL):** Gestión de usuarios, canchas, equipos, y reservas.    * **Seguridad y autenticación:** Sistema de autenticación seguro y protección de datos de los usuarios. 4. **Documentación Técnica:**    * **Manual de usuario:** Guía detallada para los usuarios sobre cómo utilizar la plataforma, tanto en la aplicación móvil como en la web.    * **Manual técnico:** Documentación para los desarrolladores sobre la arquitectura, implementación y mantenimiento de la plataforma.    * **Diagrama de arquitectura del sistema:** Muestra la estructura técnica de la plataforma, incluyendo la interacción entre los distintos módulos (móvil, web, backend, base de datos). 5. **Pruebas y Validaciones:**    * **Pruebas unitarias e integrales:** Para asegurar que cada componente del sistema funcione correctamente y que la integración entre los módulos sea efectiva.    * **Pruebas de usabilidad:** Realizadas con usuarios finales para evaluar la experiencia de usuario y realizar mejoras antes del lanzamiento final.    * **Informe de resultados de pruebas:** Documento detallando los resultados de las pruebas y las mejoras realizadas a partir del feedback recibido.   **Plazos y cronograma de entregables:**   * **Sprint 1 a Sprint 3:** Desarrollo del back-end, front-end móvil, y web. * **Sprint 4:** Pruebas y correcciones finales. * **Entrega final:** Plataforma funcional lista para su despliegue y lanzamiento. |

# **Requerimientos de alto nivel**

## **Requerimientos del producto**

|  |
| --- |
| **Aplicación Móvil (Ionic):**   * **Registro e Inicio de Sesión:** El sistema debe permitir que los usuarios se registran y accedan mediante una cuenta creada en FlashMatch o mediante autenticación externa, como Google. * **Gestión de Canchas:** La aplicación debe permitir a los usuarios consultar la disponibilidad de canchas en tiempo real, visualizar sus características y reservar canchas directamente desde la plataforma. * **Gestión de Equipos:** Los usuarios deben poder crear, gestionar y administrar equipos deportivos, asignar roles a los jugadores y visualizar las estadísticas de los equipos. * **Estadísticas de Rendimiento:** La aplicación debe generar y mostrar estadísticas detalladas de rendimiento deportivo personalizadas para cada usuario. * **Integración con el Sistema de Pagos:** El sistema debe permitir realizar pagos seguros para la reserva de canchas mediante la integración con un sistema de pagos, como **Transbank**.   **Página Web de Presentación (Angular):**   * **Página de Aterrizaje:** Debe ofrecer una descripción clara de las funcionalidades de la plataforma, permitiendo a los usuarios descargar la aplicación móvil y contactar con el equipo de soporte. * **Información sobre Funcionalidades:** La página debe explicar de manera clara las ventajas y características principales de FlashMatch, mostrando ejemplos y capturas de pantalla.   **Backend (NestJS) y Base de Datos (PostgreSQL):**   * **APIs para la Gestión de Datos:** El sistema debe contar con APIs seguras que gestionan la interacción entre el front-end y la base de datos, controlando el acceso a la información de usuarios, equipos y canchas. * **Sistema de Seguridad y Autenticación:** El sistema debe garantizar la seguridad de los datos del usuario mediante un sistema de autenticación robusto y la protección de la información personal. * **Escalabilidad:** El sistema debe ser escalable para poder manejar un gran número de usuarios concurrentes y soportar futuras expansiones, como la inclusión de nuevos deportes. |

## **Requerimientos del proyecto**

|  |
| --- |
| **Metodología de Desarrollo (Scrum):**   * El proyecto debe seguir un enfoque iterativo basado en la metodología **Scrum**, organizado en sprints de dos semanas, con entregas incrementales de valor. * El **Product Owner** y el equipo de desarrollo deben definir el backlog de producto con todas las funcionalidades clave que serán implementadas a lo largo del proyecto.   **Recursos Humanos y Roles:**   * **Jefe de Proyecto:** Responsable de la coordinación general, planificación y control de todas las actividades relacionadas con el desarrollo de la plataforma. * **Desarrolladores Full Stack:** Encargados de implementar las funcionalidades tanto en el back-end (NestJS) como en el front-end móvil (Ionic) y web (Angular). * **Tester:** Responsable de realizar las pruebas unitarias, de integración y de usabilidad para asegurar la calidad del sistema. * **Diseñador UX/UI:** Encargado de la creación y optimización de la interfaz de usuario para asegurar una experiencia atractiva e intuitiva.   **Recursos Técnicos:**   * **Herramientas de Desarrollo:** Se utilizarán **NestJS** para el backend, **Ionic** para la aplicación móvil, **Angular** para la web, y **PostgreSQL** para la base de datos. * **Herramientas de Colaboración:** Uso de **GitHub** para el control de versiones y la colaboración entre los desarrolladores, y **Postman** para pruebas de APIs. * **Infraestructura en la nube:** El proyecto utilizará servicios de hosting en la nube para asegurar la disponibilidad y escalabilidad de la plataforma.   **Plazos y Entregables:**   * El proyecto debe completarse dentro de los plazos establecidos en el cronograma, dividido en sprints, con revisiones periódicas para asegurar que los entregables estén alineados con los objetivos del proyecto. * Las entregas finales incluirán la **aplicación móvil completamente funcional**, la **página web de presentación**, la **documentación técnica** y el **informe de pruebas y validación**. |

# **Objetivos**

| **Objetivo** | **Indicador de éxito** |
| --- | --- |
| **Alcance** | |
| Implementar una plataforma móvil y web para la reserva de canchas y gestión de equipos deportivos. La plataforma debe ser funcional y accesible desde dispositivos móviles. Al menos 100 reservas gestionadas en el primer mes, con un promedio de calificación de usuarios de 4.5/5. | |
| **Cronograma (Tiempo)** | |
| Completar el desarrollo del proyecto en un total de 4 sprints durante 6 meses. Cada sprint debe ser revisado con entregables parciales y pruebas de funcionalidad. El proyecto debe estar listo para su lanzamiento en la fecha estipulada con un margen de error de dos semanas. | |
|  | |
|  | |
| **Calidad** | |
| La plataforma debe cumplir con altos estándares de calidad: funcionalidad, usabilidad, y seguridad de datos. Se espera que la plataforma sea valorada con al menos un 4.5/5 en pruebas de usuarios. Además, se garantiza un tiempo de actividad del sistema del 99% durante el primer año. | |
| **Otros** | |
| El sistema debe ser modular y fácilmente escalable, permitiendo la adición de nuevas funcionalidades como estadísticas avanzadas y la integración con otros deportes. | |

# **Premisas y restricciones**

|  |
| --- |
| **Premisas**  1. **Disponibilidad de recursos tecnológicos**: Se asume que el equipo tendrá acceso constante a las herramientas de desarrollo requeridas (GitHub, Ionic, Angular, NestJS, etc.) y que los servidores estarán disponibles para las pruebas de la aplicación. 2. **Participación activa del equipo de desarrollo**: Todos los integrantes del equipo, incluyendo el Product Owner y los desarrolladores, se comprometen a cumplir con los plazos establecidos en el cronograma, siguiendo la metodología Scrum y asistiendo a las reuniones planificadas. 3. **Disponibilidad de documentación y soporte técnico**: Se presupone que la empresa patrocinadora proporcionará el acceso a la información necesaria sobre las canchas, el proceso de reservas y la gestión de equipos. 4. **Interacción fluida con el cliente**: Se asume que habrá una comunicación regular y efectiva con el cliente para obtener retroalimentación, resolver dudas y ajustar los requisitos del proyecto según sea necesario. 5. **Estabilidad de los requisitos**: Se espera que los requisitos no sufran cambios drásticos durante los sprints, permitiendo avanzar con el desarrollo de funcionalidades críticas dentro del cronograma previsto.  **Restricciones**  1. **Presupuesto limitado**: El desarrollo del proyecto debe mantenerse dentro del presupuesto aprobado, que es de CLP 5,000,000, lo cual incluye el desarrollo de la aplicación móvil, la versión web de presentación, y los costos de infraestructura. Cualquier desviación debe ser mínima y justificada. 2. **Tiempo de desarrollo**: El proyecto debe completarse en un máximo de 6 meses, distribuidos en 4 sprints, lo que limita la cantidad de funcionalidades que se pueden desarrollar durante ese tiempo. Las funcionalidades adicionales o mejoras estarán sujetas a disponibilidad posterior. 3. **Recursos humanos**: El equipo de desarrollo está compuesto por tres personas, lo que implica que cada miembro debe asumir múltiples roles (desarrollo back-end, front-end, pruebas, etc.), limitando la capacidad para implementar todas las funcionalidades al mismo tiempo. 4. **Calidad de la aplicación**: La aplicación debe cumplir con estándares de seguridad y privacidad de datos, particularmente en el manejo de la información de los usuarios, como credenciales y datos personales. Esto añade restricciones al tiempo de desarrollo, ya que implica la implementación de mecanismos de seguridad adecuados. |

# **Riesgos iniciales de alto nivel**

|  |
| --- |
| **Riesgos Iniciales de Alto Nivel**  1. **Retrasos en el cronograma** *Riesgo:* Existe la posibilidad de que los plazos establecidos no se cumplan debido a problemas imprevistos en el desarrollo, errores en las funcionalidades críticas o retrasos en la retroalimentación del cliente. *Impacto:* Alto – Retrasar el cronograma podría afectar la entrega final y comprometer el alcance del proyecto. *Plan de mitigación:* Se implementará un seguimiento riguroso de los sprints mediante la metodología Scrum, incluyendo revisiones semanales para identificar bloqueos tempranos y ajustar las tareas o redistribuir la carga de trabajo si es necesario. 2. **Problemas de integración entre el front-end y el back-end** *Riesgo:* Al ser el primer proyecto en el que se trabaja con la combinación de tecnologías como Ionic, Angular y NestJS, podría haber incompatibilidades o dificultades en la integración de los módulos. *Impacto:* Medio – Los problemas de integración pueden causar que las funcionalidades no operen correctamente, lo que afectaría la experiencia del usuario. *Plan de mitigación:* Realizar pruebas de integración continuas desde el Sprint 2, asegurando que los componentes del back-end y front-end interactúen sin problemas desde una etapa temprana del desarrollo. 3. **Falta de claridad en los requisitos del cliente** *Riesgo:* Los requisitos del cliente podrían no estar bien definidos desde el inicio o podrían cambiar durante el desarrollo del proyecto, lo que causaría confusión en el equipo y cambios significativos en el alcance. *Impacto:* Alto – Cambios en los requisitos podrían generar un trabajo adicional y desalineación con el cronograma. *Plan de mitigación:* Mantener reuniones frecuentes con el cliente y revisar el Product Backlog de manera constante para asegurar que los requisitos estén bien definidos y aprobados antes de proceder con el desarrollo. 4. **Disponibilidad limitada del equipo de desarrollo** *Riesgo:* Dado que el equipo es pequeño y cada miembro cumple varios roles, la falta de disponibilidad por problemas de salud o compromisos externos puede retrasar el avance. *Impacto:* Alto – La baja disponibilidad de un miembro puede ralentizar todo el proceso de desarrollo, dada la dependencia de tareas específicas. *Plan de mitigación:* Establecer prioridades claras dentro de los sprints y asegurar que todas las tareas tengan al menos un plan de contingencia, distribuyendo tareas críticas entre los miembros disponibles. 5. **Fallas en la seguridad de la aplicación** *Riesgo:* La aplicación maneja datos sensibles de los usuarios (como contraseñas y datos personales), por lo que un fallo en la seguridad podría derivar en la filtración de datos. *Impacto:* Muy alto – Una brecha de seguridad podría dañar la reputación del proyecto y causar problemas legales. *Plan de mitigación:* Implementar medidas de seguridad estrictas, como autenticación segura y cifrado de datos, desde las primeras fases del desarrollo. Realizar pruebas de vulnerabilidad periódicas y revisiones de código centradas en la seguridad. 6. **Falta de compatibilidad con diferentes dispositivos y sistemas operativos** *Riesgo:* La aplicación móvil debe ser compatible tanto con dispositivos Android como iOS, pero podrían surgir problemas de compatibilidad debido a las diferentes versiones de sistemas operativos. *Impacto:* Medio – La falta de compatibilidad podría limitar el número de usuarios que pueden utilizar la aplicación y afectar la adopción del producto. *Plan de mitigación:* Realizar pruebas en múltiples dispositivos y versiones de sistemas operativos desde las primeras etapas del desarrollo, y ajustar el código según los problemas detectados. |

# **Cronograma de hitos principales**

|  |  |
| --- | --- |
| **Hito** | **Fecha tope** |
| Creación del Acta de Constitución del Proyecto | 15 de julio de 2024 |
| Reunión de Kick-Off | 20 de julio de 2024 |
| Definición del Product Backlog | 25 de julio de 2024 |
| Preparación del Sprint 0 | 10 de agosto de 2024 |
| Revisión del Sprint 0 | 30 de agosto de 2024 |
| Sprint 1: Desarrollo del Back-End | 25 de septiembre de 2024 |
| Sprint 2: Desarrollo del Front-End | 2 de octubre de 2024 |
| Sprint 3: Integración de Componentes | 23 de octubre de 2024 |
| Sprint 4: Pruebas y Validación | 30 de octubre de 2024 |
| Implementación de IA (Sprint 5) | 15 de noviembre de 2024 |

# **Presupuesto estimado**

|  |
| --- |
| El presupuesto estimado para el proyecto **FlashMatch** contempla varios aspectos clave para asegurar el cumplimiento de los objetivos del proyecto, con un enfoque en los costos de recursos humanos, herramientas, infraestructura y otros gastos operativos.  En cuanto a los **recursos humanos**, se incluye el salario de los desarrolladores Full Stack, como Daniel González y Carlos Guzmán, que trabajarán en el proyecto por aproximadamente dos meses, junto con el Product Owner, Rodrigo Berrios. Estos costos representan una parte significativa del presupuesto total.  Las **herramientas y licencias** necesarias incluyen plataformas como GitHub, que ofrece funciones avanzadas para la colaboración y el control de versiones del código, y tecnologías como Ionic y Angular para el desarrollo de la aplicación móvil y web, respectivamente. Además, se utilizará el framework NestJS para el backend, que no implica un costo adicional.  En términos de **infraestructura y servicios**, se consideran los costos de servidores en la nube (AWS), los cuales serán fundamentales para alojar y ejecutar la aplicación, así como el servicio de bases de datos para almacenar la información de usuarios y reservas. También se ha considerado la integración con Transbank para habilitar pagos en línea, que generará una pequeña comisión por cada transacción.  El **costo total estimado** del proyecto es de **aproximadamente 8,168 USD**, cubriendo todos los aspectos mencionados anteriormente. Este presupuesto está sujeto a ajustes dependiendo de la evolución del proyecto y los requerimientos adicionales que puedan surgir. |

# **Indicadores de Éxito**

|  |
| --- |
| **Descripción**: Los indicadores de éxito permiten evaluar de manera objetiva el progreso del proyecto y verificar el cumplimiento de los objetivos establecidos, asegurando que el equipo mantenga un alto nivel de calidad en cada iteración. Estos indicadores permiten ajustes continuos para alinear el proyecto con los requerimientos y expectativas del cliente y del equipo.  **Indicador 1: Satisfacción del Cliente**   * **Objetivo**: Asegurar que el cliente esté satisfecho con el desarrollo y las entregas del proyecto en cada iteración. * **Método de Evaluación**: Al final de cada sprint, el equipo realiza una encuesta al cliente, evaluando aspectos como:   + Cumplimiento de expectativas.   + Calidad de la entrega.   + Comunicación y transparencia en el desarrollo. * **Escala de Evaluación**: Escala de 1 a 5, donde 5 es “muy satisfecho” y 1 es “muy insatisfecho”. * **Meta**: Mantener un puntaje promedio de satisfacción igual o superior a **4/5**. * **Acciones en caso de desviación**: Si el puntaje es inferior a 4 en un sprint, se realiza una reunión de ajuste con el cliente para identificar y corregir áreas de mejora en el siguiente sprint.   **Indicador 2: Cumplimiento de Plazos**   * **Objetivo**: Evaluar la efectividad del equipo en cumplir con el cronograma y los plazos establecidos para cada tarea y entrega de sprint. * **Método de Evaluación**: Al final de cada sprint, se calcula el porcentaje de tareas completadas en relación con el total de tareas planificadas. * **Fórmula de Cálculo**:   Cumplimiento de Plazos =(Tareas Completadas/Tareas Planificadas)×100   * **Meta**: Alcanzar un cumplimiento de **90% o más** de las tareas en los tiempos programados para cada sprint. * **Acciones en caso de desviación**: En caso de cumplir menos del 90%, se realiza una evaluación de impedimentos y un ajuste en la planificación para mejorar en el próximo sprint.   **Indicador 3: Tasa de Incidencias**   * **Objetivo**: Minimizar la cantidad de errores críticos en cada entrega de sprint para asegurar la calidad del código y la estabilidad de la aplicación. * **Método de Evaluación**: Número de errores críticos identificados en cada sprint mediante pruebas de calidad (QA) y feedback del equipo. * **Categorización de Incidencias**:   + *Críticas*: Errores que afectan directamente la funcionalidad principal y bloquean el uso.   + *Menores*: Errores que no afectan la funcionalidad crítica pero requieren solución. * **Meta**: Reducir la cantidad de incidencias críticas en cada iteración mediante la implementación de mejoras continuas. * **Acciones en caso de desviación**: Si aumenta el número de incidencias críticas en un sprint, se introduce un sprint de mantenimiento o se reasignan recursos para mejorar la calidad del código en las áreas afectadas.   **Indicador 4: Tiempo de Resolución de Incidencias**   * **Objetivo**: Monitorear la rapidez con la que el equipo responde y resuelve errores críticos para mantener una entrega de alta calidad en cada sprint. * **Método de Evaluación**: Medir el tiempo promedio desde la detección de la incidencia crítica hasta su resolución. * **Fórmula de Cálculo**:   Tiempo de Resolución=Tiempo Total de Resolución/Número de Incidencias   * **Meta**: Resolver incidencias críticas en menos de **24 horas**. * **Acciones en caso de desviación**: Si el tiempo promedio de resolución supera las 24 horas, se introduce un análisis de los flujos de trabajo para mejorar la asignación de tareas y la prioridad de resolución.   **Indicador 5: Cobertura de Pruebas**   * **Objetivo**: Asegurar que la mayor parte del código esté cubierto por pruebas unitarias e integradas para evitar errores en la producción. * **Método de Evaluación**: Medir el porcentaje de líneas de código cubiertas por pruebas al final de cada sprint. * **Herramientas de Pruebas**: Utilización de herramientas de cobertura de pruebas, como Jest para pruebas unitarias en el backend y Jasmine/Karma para el frontend. * **Fórmula de Cálculo**: Cobertura de Pruebas=(Líneas Cubiertas por Pruebas/Total de Líneas de Código)×100 * **Meta**: Alcanzar una cobertura de al menos **80%** en cada sprint para asegurar la calidad del código. * **Acciones en caso de desviación**: Si la cobertura es menor al 80%, se reasignan recursos para aumentar la cobertura de pruebas en el próximo sprint.   **Indicador 6: Eficiencia del Equipo (Opcional)**   * **Objetivo**: Medir la productividad del equipo en relación con el tiempo y los recursos invertidos en el proyecto. * **Método de Evaluación**: Comparar el tiempo estimado de cada tarea con el tiempo real registrado y analizar la diferencia en cada sprint. * **Fórmula de Cálculo**: Eficiencia=(Tiempo Estimado/Tiempo Real)×100 * **Meta**: Mantener una eficiencia promedio superior al **85%**. * **Acciones en caso de desviación**: En caso de baja eficiencia, se evalúan los métodos de estimación y se ajustan para reflejar de mejor manera el tiempo requerido en sprints futuros. |

# **Lista de Interesados (stakeholders)**

| **Nombre** | **Cargo** | **Departamento / División** |
| --- | --- | --- |
| Rodrigo Berrios | Product Owner / Desarrollador FullStack | Departamento de Desarrollo |
| Daniel González | Desarrollador FullStack | Departamento de Desarrollo |
| Carlos Guzmán | Scrum Master / Desarrollador FullStack | Departamento de Desarrollo |
| Fabián Alcántara | Docente Supervisor | Escuela de Informática y Telecomunicaciones |
| FlashMatch | Cliente / Patrocinador | Departamento de Operaciones |
| Directores de Canchas | Encargados de gestión de reservas | Departamentos de Operaciones Canchas |
| Transbank | Proveedor de Servicios de Pago |  |
| Usuarios (Jugadores y Equipos) | Usuarios Finales | N/A |
| Administradores de Sistema | Administradores | Departamento de TI |

# **Requisitos de aprobación del proyecto**

|  |
| --- |
| **Cumplimiento del Alcance del Proyecto**:   * El proyecto debe cumplir con todas las funcionalidades definidas en el alcance original y ajustado del proyecto, incluyendo la aplicación móvil para la reserva de canchas, la interfaz web de presentación, y la gestión de usuarios. * Las funcionalidades postergadas (reconocimiento facial y mensajería) deben quedar documentadas y planificadas para futuras versiones.   **Calidad del Producto**:   * La aplicación móvil y la interfaz web deben pasar por pruebas exhaustivas que aseguren un funcionamiento óptimo, sin errores críticos que afecten la experiencia del usuario final. * Las pruebas de integración entre el front-end y el back-end deben demostrar la correcta comunicación entre ambos sistemas y un rendimiento estable. * La interfaz debe ser intuitiva y garantizar una buena experiencia de usuario (UX), tanto en la versión móvil como en la versión web.   **Cumplimiento de Plazos (Cronograma)**:   * El proyecto debe cumplir con los hitos establecidos en el cronograma del proyecto. Cada sprint debe ser completado a tiempo y revisado al finalizar, asegurando que los entregables cumplan con los criterios de éxito definidos. * La entrega final del proyecto debe ser antes del 15 de noviembre de 2024, con una aplicación funcional y todos los documentos técnicos requeridos.   **Cumplimiento del Presupuesto**:   * El proyecto debe mantenerse dentro del presupuesto estimado y planificado, considerando los recursos humanos, tecnológicos y financieros asignados. * Cualquier desvío presupuestario debe ser aprobado por los patrocinadores del proyecto y documentado adecuadamente.   **Satisfacción del Cliente y Usuarios**:   * El proyecto será aprobado tras recibir una validación formal por parte de FlashMatch, el cliente/patrocinador principal, asegurando que se cumplan sus expectativas y necesidades. * Se debe realizar una encuesta de satisfacción a un grupo piloto de usuarios para evaluar la usabilidad y efectividad del sistema.   **Documentación Completa y Aprobada**:   * Toda la documentación técnica, incluyendo el acta de constitución, product backlog, y documentación de pruebas y validación, debe estar completa y ser aprobada por el equipo directivo y el cliente. * Se debe entregar un informe final que resuma los resultados del proyecto, incluyendo las lecciones aprendidas, las decisiones tomadas y el cumplimiento de los objetivos.   **Conformidad Legal y Normativa**:   * El sistema debe cumplir con todas las normativas legales vigentes, especialmente en relación con la protección de datos personales de los usuarios, según la Ley de Protección de Datos Personales (Ley 19.628 en Chile). * Las integraciones de pagos deben cumplir con los requisitos de seguridad de Transbank y los estándares de la industria (PCI DSS). |

# **Asignación del gerente de proyecto y nivel de autoridad**

## **Gerente de proyecto**

| **Nombre** | **Cargo** | **Departamento / División** | **Rama ejecutiva (Vicepresidencia)** |
| --- | --- | --- | --- |
| Carlos Guzmán Oyarzún | Scrum Master / Desarrollador FullStack | Escuela de Informática y Telecomunicaciones, Capstone Duoc UC | Dirección Académica y Proyectos Especiales, Duoc UC |

# **Niveles de autoridad**

|  |  |
| --- | --- |
| **Área de autoridad** | **Descripción del nivel de autoridad** |
| Decisiones de personal (Staffing) | El Gerente de Proyecto tiene la autoridad para asignar roles y responsabilidades dentro del equipo de desarrollo, incluyendo la reasignación de tareas en función de la disponibilidad y las habilidades del equipo. En caso de que se requiera personal adicional, se debe escalar la solicitud al patrocinador principal. |
| Gestión de presupuesto y de sus variaciones | El Gerente de Proyecto tiene control directo sobre la gestión del presupuesto asignado para el desarrollo de la aplicación, asegurando que se mantenga dentro de los límites establecidos. Cualquier variación presupuestaria significativa debe ser aprobada por los patrocinadores antes de la implementación. |
| Decisiones técnicas | Tiene plena autoridad para tomar decisiones técnicas relativas a la arquitectura del sistema, las tecnologías a utilizar (Ionic, Angular, NestJS), y las integraciones necesarias. Decisiones técnicas de gran envergadura, como cambios en la plataforma tecnológica o la incorporación de nuevos módulos, requieren consulta con el Product Owner. |
| Resolución de conflictos | El Gerente de Proyecto es el responsable de mediar en cualquier conflicto que surja dentro del equipo. En caso de que el conflicto no pueda ser resuelto internamente, se debe escalar al patrocinador o al Product Owner para su resolución. |

# **Personal y recursos preasignados**

| Recurso | Departamento / División | Rama ejecutiva (Vicepresidencia) |
| --- | --- | --- |
| Daniel González | Desarrollo FullStack | Escuela de Informática y Telecomunicaciones, Duoc UC |
| Rodrigo Berrios | Product Owner / Desarrollador FullStack | Escuela de Informática y Telecomunicaciones, Duoc UC |
| Carlos Guzmán | Scrum Master / Desarrollador FullStack | Escuela de Informática y Telecomunicaciones, Duoc UC |
| Servidor Backend | Infraestructura y Servicios Tecnológicos | Dirección de Tecnología, Duoc UC |
| Repositorios GitHub | Plataforma de Gestión de Código y Control de Versiones | Dirección de Tecnología, Duoc UC |
| Entorno de desarrollo local (Ionic, Angular, NestJS) | Desarrollo Tecnológico | Dirección de Tecnología, Duoc UC |
| Hosting y base de datos | Infraestructura y Servicios de Nube | Dirección de Tecnología, Duoc UC |