**Documento de requerimientos de software**

***Proyecto de Gestión de ayudantías - StudIA***

***Fecha: [05/09/2025]***

**Tabla de contenido**

Historial de Versiones 3

Información del Proyecto 3

Aprobaciones 3

1. Propósito 4

2. Alcance del producto / Software 4

3. Referencias 4

4. Funcionalidades del producto 5

5. Clases y características de usuarios 5

6. Entorno operativo 5

7. Requerimientos funcionales 6

9.1. (Nombre de la funcionalidad 1) 6

9.2. (Nombre de la funcionalidad 2) 7

9.3. (Nombre de la funcionalidad N) 7

8. Reglas de negocio 8

9. Requerimientos de interfaces externas 9

9.1. Interfaces de usuario 9

9.2. Interfaces de hardware 9

9.3. Interfaces de software 9

9.4. Interfaces de comunicación 9

10. Requerimientos no funcionales 10

11. Otros requerimientos 11

12. Glosario 12

**Historial de Versiones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| 30-08-2025 | 0.1 | Eduardo Álvarez | DuocUC | Creación inicial del documento |
| 30-08-2025 | 0.2 | Felipe Silva | DuocUC | Revisión y complementación inicial |
|  |  |  |  |  |
|  |  |  |  |  |

**Información del Proyecto**

|  |  |
| --- | --- |
| Empresa / Organización | DuocUC |
| Proyecto | Sistema de Gestión de Ayudantías |
| Fecha de preparación | 30-08-2025 |
| Cliente | DuocUC (Coordinación Académica) |
| Patrocinador principal | DuocUC |
| Gerente / Líder de Proyecto | Eduardo Álvarez |
| Gerente / Líder de Análisis de negocio y requerimientos | Felipe Silva |

**Aprobaciones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
| Eduardo Álvarez | Líder de Proyecto | DuocUC – Estudiante | 30-08-2025 |  |
| Felipe Silva | Analista de requerimientos | DuocUC – Estudiante | 30-08-2025 |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. **Propósito**

El presente documento tiene como propósito especificar los requerimientos del sistema denominado **StudIA – Sistema de Gestión de Ayudantías** (Versión 1.0, Release inicial).

El software surge como respuesta a una problemática detectada en **DuocUC**: la alta tasa de deserción y el bajo rendimiento académico en asignaturas críticas, producto de la falta de herramientas efectivas para coordinar y gestionar ayudantías.

El sistema buscará:

* Facilitar el acceso de los estudiantes a instancias de apoyo académico.
* Estandarizar y optimizar la gestión de ayudantías, horarios y cupos.
* Brindar una herramienta centralizada para estudiantes, ayudantes y docentes.
* Contribuir a la reducción de la deserción académica y a la mejora del rendimiento.
* Permitir que estudiantes destacados puedan generar ingresos adicionales a través de su rol como ayudantes.

Este documento cubre la **totalidad del software en su primera versión**, incluyendo los procesos de inscripción, asignación y administración de ayudantías. Funcionalidades futuras como integración con sistemas institucionales externos o módulos avanzados de inteligencia artificial quedarán fuera del alcance de la versión inicial, pero podrán ser consideradas en releases posteriores.

1. **Alcance del producto / Software**

2.1 Propósito u objetivo general

El sistema StudIA – Sistema de Gestión de Ayudantías tiene como objetivo principal mejorar la eficiencia y efectividad del proceso de gestión de ayudantías en DuocUC, brindando una plataforma que permita a estudiantes acceder fácilmente a instancias de apoyo académico, a los ayudantes gestionar su disponibilidad y a los docentes/coordinadores supervisar y administrar el proceso de manera centralizada.

2.2 Beneficios para el área de negocio y la organización

* Académicos: Disminución de la deserción estudiantil y aumento en la tasa de aprobación de asignaturas con alto nivel de dificultad.
* Operacionales: Optimización del tiempo y recursos destinados a la coordinación de ayudantías, reduciendo errores manuales y duplicidad de información.
* Económicos: Posibilidad de que estudiantes destacados generen ingresos adicionales a través de su participación como ayudantes.
* Institucionales: Fortalecimiento del compromiso de DuocUC con la innovación educativa y la mejora continua de la experiencia de sus estudiantes.

2.3 Objetivos y metas

* Objetivo general:  
  Implementar una plataforma tecnológica que apoye la gestión de ayudantías, alineada con las estrategias de innovación, eficiencia operativa y apoyo al estudiante promovidas por DuocUC.

Objetivos específicos:

1. Facilitar a los estudiantes la inscripción y acceso a ayudantías en asignaturas críticas.
2. Estandarizar la gestión de horarios, cupos y disponibilidad de ayudantes.
3. Proporcionar herramientas de seguimiento a docentes y coordinadores para apoyar la toma de decisiones.
4. Reducir la deserción académica mediante un apoyo más oportuno y eficiente.
5. Contribuir al desarrollo profesional de estudiantes destacados al permitirles ejercer como ayudantes remunerados.

1. **Referencias**

A continuación, se listan los documentos y fuentes que complementan y respaldan la presente especificación de requerimientos del sistema StudIA – Sistema de Gestión de Ayudantías:

Informe Final Proyecto APT

* Autores: Eduardo Álvarez B., Felipe Silva
* Profesor guía: Fernando Herrera
* Versión / Fecha: 30-08-2025
* Ubicación: <https://drive.google.com/drive/folders/1Kp4hky0b-zVrqm3LeaMtAPwMQffJ1RpV>

Prototipos de la Plataforma

* Autores: Equipo de proyecto APT (Eduardo Álvarez B., Felipe Silva)
* Versión: Borrador inicial
* Fecha: 30-08-2025
* Ubicación: <https://drive.google.com/drive/folders/1Kp4hky0b-zVrqm3LeaMtAPwMQffJ1RpV>

Metodología Scrum – Guía oficial

* Autores: Ken Schwaber, Jeff Sutherland
* Versión: Guía de Scrum 2020
* Ubicación: https://scrumguides.org

1. **Funcionalidades del producto**

El sistema StudIA – Sistema de Gestión de Ayudantías (Versión 1.0) contempla las siguientes funcionalidades principales, desarrolladas bajo una base de datos relacional SQL, con módulos diferenciados según perfil de usuario y capacidades de integración con otros sistemas:

4.1 Módulo Administrador

* Gestión de usuarios (crear, modificar, eliminar, asignar roles).
* Administración de ayudantías (CRUD completo de asignaturas, horarios, cupos y ayudantes).
* Gestión de integraciones con otros sistemas (ej: portal de pagos, geolocalización de sedes, APIs externas).

4.2 Módulo Usuario (Estudiante)

* Registro e inicio de sesión.
* Búsqueda de ayudantías por asignatura, sede, horario o modalidad.
* Inscripción y cancelación en ayudantías.
* Recepción de notificaciones y recordatorios.
* Visualización de historial de ayudantías inscritas.

4.3 Módulo Supervisor (Docente / Coordinador)

* Consultas y dashboards sobre la asistencia y participación en ayudantías.
* Generación de reportes (por asignatura, estudiante, ayudante o período).
* Monitoreo de cupos y rendimiento de ayudantías.

4.4 Funcionalidades transversales

* Base de datos SQL relacional para almacenamiento y gestión de la información.
* Integración con sistemas externos (ej: APIs de pagos, mapas de sedes, otros portales institucionales).
* Desarrollo web y móvil (app), garantizando acceso multiplataforma.
* Diseño responsivo para adaptación a diferentes tamaños de pantalla y dispositivos.
* Lenguajes de programación utilizados en la formación académica (alineado a los ramos de la carrera).

1. **Clases y características de usuarios**

El sistema StudIA – Sistema de Gestión de Ayudantías será utilizado por diferentes perfiles de usuario, cada uno con características particulares en cuanto a frecuencia de uso, nivel de acceso y funcionalidades relevantes.

5.1 Administrador

* Descripción: Usuario con permisos completos para la gestión de la plataforma. Puede ser personal designado por DuocUC para administrar la información global del sistema.
* Frecuencia de uso: Media, en tareas de configuración y mantenimiento.
* Privilegios de seguridad: Acceso total (CRUD sobre usuarios, ayudantías y configuraciones).
* Nivel de experiencia: Medio/alto, se espera que tenga conocimientos básicos en sistemas de gestión.

Funcionalidades relevantes:

* Gestión de usuarios y roles.
* Administración de ayudantías (crear, modificar, eliminar).
* Integraciones con sistemas externos.

5.2 Usuario Estudiante

* Descripción: Estudiante de DuocUC que busca inscribirse en ayudantías para reforzar sus asignaturas.
* Frecuencia de uso: Alta, especialmente en periodos de inscripción o previo a evaluaciones.
* Privilegios de seguridad: Acceso restringido a su perfil personal y a la inscripción en ayudantías.
* Nivel de experiencia: Bajo/medio, se espera un uso intuitivo y accesible.

Funcionalidades relevantes:

* Registro e inicio de sesión.
* Inscripción y cancelación de ayudantías.
* Visualización de historial.
* Notificaciones y recordatorios.

5.3 Usuario Ayudante

* Descripción: Estudiante destacado que ofrece ayudantías en determinadas asignaturas.
* Frecuencia de uso: Media/alta, según la cantidad de ayudantías que gestione.
* Privilegios de seguridad: Acceso restringido a la creación y administración de sus ayudantías.
* Nivel de experiencia: Medio, debe manejar funcionalidades de planificación y comunicación.

Funcionalidades relevantes:

* Registro de disponibilidad y horarios.
* Creación de sesiones de ayudantía.
* Comunicación con estudiantes inscritos.
* Control de asistencia.

5.4 Supervisor (Docente / Coordinador)

* Descripción: Docente o coordinador académico encargado de monitorear la correcta ejecución de las ayudantías.
* Frecuencia de uso: Baja/media, con acceso en momentos puntuales para revisión y seguimiento.
* Privilegios de seguridad: Acceso a consultas, reportes y dashboards. No realiza modificaciones directas sobre el sistema.
* Nivel de experiencia: Medio/alto, con orientación al análisis de datos y gestión académica.

Funcionalidades relevantes:

* Visualización de reportes.
* Consultas sobre inscripciones y asistencia.
* Métricas académicas para la toma de decisiones.

1. **Entorno operativo**

El sistema StudIA – Sistema de Gestión de Ayudantías deberá funcionar en un entorno tecnológico compatible con la infraestructura académica de DuocUC y accesible a estudiantes, ayudantes, docentes y administradores desde distintas plataformas.

6.1 Plataforma de hardware

* Servidores: El sistema deberá estar alojado en servidores locales o en la nube (ej. AWS, Azure, GCP), con capacidad suficiente para manejar la carga de usuarios simultáneos prevista.
* Clientes:
* Equipos de escritorio y notebooks: acceso vía navegador web.
* Dispositivos móviles: smartphones y tablets con sistemas operativos Android e iOS.

6.2 Plataforma de software

* Sistema operativo servidor: Linux (Ubuntu Server o equivalente) o Windows Server (según decisión institucional).
* Sistema operativo cliente: Windows, macOS, Android, iOS.
* Base de datos: SQL relacional (ej. MySQL, PostgreSQL o SQL Server).
* Navegadores soportados: Google Chrome, Microsoft Edge, Mozilla Firefox, Safari.
* Frameworks y lenguajes de desarrollo:
* Frontend: HTML5, CSS3, JavaScript (framework sugerido: React).
* Backend: Node.js, Java o .NET (según disponibilidad en los ramos cursados).
* Móvil: Desarrollo nativo (Java/Kotlin para Android, Swift para iOS) o multiplataforma (ej. React Native, Flutter).

6.3 Conectividad e integración

* Red: Acceso a Internet estable, tanto en campus como de forma remota.
* Integraciones: Capacidad de interoperar con sistemas externos mediante APIs REST, XML o JSON (ej: portal de pagos, mapas de sedes, servicios institucionales).
* 9.4 Seguridad y acceso
* Autenticación: sistema de login con credenciales únicas por usuario.
* Seguridad de datos: cifrado en tránsito (HTTPS/TLS) y cifrado de información sensible en base de datos.
* Roles y permisos: definidos según perfiles de usuario (Administrador, Estudiante, Ayudante, Supervisor).

1. **Requerimientos funcionales**

Se adjunta planilla de matriz de trazabilidad de requisitos.

1. **Reglas de negocio**
2. Gestión de roles

* Todo usuario debe pertenecer a un perfil único (Administrador, Estudiante, Ayudante, Supervisor).
* Los permisos están determinados por el rol y no pueden modificarse sin intervención del Administrador.

1. Cupos y disponibilidad

* Una ayudantía no puede superar el número de cupos definidos en su creación.
* Un estudiante no puede inscribirse en más de una ayudantía si existe conflicto horario.
* El ayudante debe definir su disponibilidad antes de crear nuevas ayudantías.

1. Asistencia

* La asistencia debe registrarse únicamente por el ayudante responsable de la sesión.
* La asistencia no puede modificarse una vez cerrado el acta de la sesión.

1. Integridad de datos

* Todos los usuarios deben autenticarse antes de acceder al sistema.
* Contraseñas deben almacenarse en formato cifrado (hash).
* Los datos sensibles deben cumplir normativa vigente de protección de datos personales.

1. Notificaciones

* Toda inscripción, cancelación o modificación de una ayudantía debe generar una notificación al estudiante involucrado.
* Las notificaciones deben enviarse a través de la aplicación y, opcionalmente, por correo electrónico.

1. Integraciones externas

* El sistema solo podrá integrarse con servicios externos validados por DuocUC (ej. portal de pagos institucional).
* Las integraciones deben implementarse a través de APIs seguras (REST/JSON o XML) con autenticación.

1. **Requerimientos de interfaces externas**

9. Requerimientos de Interfaces Externas

9.1 Interfaces de Usuario

El sistema StudIA – Sistema de Gestión de Ayudantías contará con interfaces web y móviles, diseñadas bajo principios de usabilidad y consistencia gráfica.

* Clasificación por áreas:
  1. Módulo Estudiante → inscripción, historial, notificaciones.
  2. Módulo Ayudante → creación de ayudantías, disponibilidad, control de asistencia.
  3. Módulo Administrador → gestión de usuarios y sistema.
  4. Módulo Supervisor → consultas y reportes.
* Estándares de GUI:
  1. Diseño responsivo (adaptación a distintos tamaños de pantalla).
  2. Estilo visual consistente: colores institucionales de DuocUC, tipografía clara y accesible.
  3. Menús de navegación superiores o laterales según dispositivo.
  4. Botones de acción principales destacados (ej: “Inscribirse”, “Guardar”, “Cancelar”).
  5. Iconografía estándar (ej: lupa para búsqueda, campana para notificaciones).
* Guías de estilo:
  1. Todas las pantallas incluirán menú de navegación, área principal de contenido y barra de notificaciones.
  2. Formularios con validaciones en tiempo real (ej: correos válidos, campos obligatorios).
  3. Accesibilidad: contraste adecuado, textos alternativos en imágenes, soporte para lectores de pantalla.

9.2 Interfaces de Hardware

El sistema debe poder ejecutarse en los siguientes dispositivos:

* Computadores de escritorio y notebooks (Windows, macOS, Linux) mediante navegador web.
* Dispositivos móviles (Android e iOS) mediante aplicación nativa o híbrida.
* Periféricos opcionales:
  1. Impresoras para exportar reportes en PDF/XLS.
* Protocolos de comunicación soportados: Wi-Fi, Ethernet y redes móviles (4G/5G).
* Interacciones hardware-software:
  1. Procesamiento en servidor (backend + base de datos).
  2. Visualización en cliente (web/móvil).

9.3 Interfaces de Software

* Bases de datos: Relacional SQL (MySQL, PostgreSQL o SQL Server).
* Sistema operativo servidor: Linux (Ubuntu Server recomendado) o Windows Server.
* Frameworks y librerías sugeridas:
  + Frontend: React, Bootstrap/TailwindCSS.
  + Backend: Node.js con Express / Java con Spring Boot / .NET Core (dependiendo de lo disponible en DuocUC).
  + Móvil: React Native o Flutter (alternativamente nativo en Kotlin/Swift).
* Integraciones externas:
* Portal de pagos institucional (mediante API REST o SOAP).
* Servicios de mapas/ubicación (ej: Google Maps API).
* Servicios de mensajería (SMTP para correos, notificaciones push en móviles).
* Compatibilidad:
* Navegadores: Chrome, Edge, Firefox, Safari (últimas versiones estables).
* Clientes móviles Android 10+ e iOS 13+.

9.4 Interfaces de Comunicación

El sistema utilizará protocolos de comunicación y seguridad estandarizados:

* Protocolos web: HTTP/HTTPS (todas las comunicaciones cifradas con TLS 1.2+).
* Servicios de mensajería:
* Notificaciones in-app.
* Notificaciones vía correo electrónico (SMTP con autenticación segura).
* Notificaciones push en dispositivos móviles.
* Formatos de mensajería: JSON y XML para intercambio de datos.
* Estándares de comunicación: REST API como principal mecanismo de interoperabilidad.
* Seguridad en comunicaciones:
* Autenticación mediante tokens JWT.
* Políticas de caducidad de sesión.
* Logs de auditoría para accesos y operaciones críticas.

1. **Requerimientos no funcionales**

10.1 Rendimiento

* RNF-1: El sistema debe permitir al menos 200 usuarios concurrentes sin degradar su rendimiento.
* RNF-2: El tiempo de respuesta en consultas críticas (ej. búsqueda de ayudantías) no debe superar los 2 segundos bajo carga normal.
* RNF-3: Las operaciones de inscripción deben confirmarse en un tiempo máximo de 3 segundos.

10.2 Seguridad

* RNF-4: Todas las comunicaciones deben estar cifradas mediante HTTPS/TLS 1.2 o superior.
* RNF-5: Las contraseñas deben almacenarse en formato hash con sal (ej. bcrypt, Argon2).
* RNF-6: El sistema debe implementar roles y permisos para controlar accesos.
* RNF-7: Debe registrarse un log de auditoría con accesos, modificaciones y errores críticos.
* RNF-8: Cumplimiento con la Ley 19.628 de protección de datos personales (Chile).

10.3 Disponibilidad y Confiabilidad

* RNF-9: El sistema debe garantizar una disponibilidad mínima del 99,5% mensual (si se aloja en la nube).
* RNF-10: El sistema debe ser tolerante a fallos en caso de caída de un servidor (ej: uso de respaldos automáticos).
* RNF-11: Se deben realizar copias de seguridad automáticas de la base de datos al menos una vez al día.

10.4 Usabilidad

* RNF-12: La interfaz debe ser responsiva, accesible desde navegadores y dispositivos móviles.
* RNF-13: El sistema debe ser intuitivo, requiriendo un máximo de 2 horas de capacitación para un usuario nuevo.
* RNF-14: Cumplimiento con estándares de accesibilidad web (WCAG 2.1 AA).

10.5 Mantenibilidad y Escalabilidad

* RNF-15: El código debe estar documentado con estándares de la institución (comentarios, manual técnico).
* RNF-16: El sistema debe permitir la integración futura con nuevos módulos mediante APIs.
* RNF-17: Las actualizaciones deben poder aplicarse sin interrumpir el servicio más de 30 minutos.

10.6 Portabilidad

* RNF-18: El sistema debe ser compatible con los navegadores Chrome, Edge, Firefox y Safari (últimas versiones).
* RNF-19: La aplicación móvil debe ser compatible con Android 10+ e iOS 13+.
* RNF-20: La base de datos debe poder migrarse entre motores SQL compatibles (MySQL, PostgreSQL, SQL Server).

1. **Otros requerimientos**

11.1 Requerimientos de Bases de Datos

* OR-1: El sistema debe utilizar una base de datos relacional SQL (MySQL, PostgreSQL o SQL Server).
* OR-2: La base de datos debe garantizar integridad referencial mediante claves primarias y foráneas.
* OR-3: Debe existir un mecanismo de respaldos automáticos diarios y restauración.
* OR-4: Las consultas frecuentes (ej. búsqueda de ayudantías) deben optimizarse mediante índices.

11.2 Requerimientos de Internacionalización y Localización

* OR-5: El sistema debe estar inicialmente en idioma español (Chile).
* OR-6: Debe permitir la incorporación futura de otros idiomas (ej. inglés) sin rediseñar la arquitectura.
* OR-7: El sistema debe manejar formatos locales (fecha DD/MM/AAAA, moneda en CLP).

11.3 Requerimientos Legales y Normativos

* OR-8: El sistema debe cumplir con la Ley 19.628 de protección de datos personales en Chile.
* OR-9: En caso de integrarse con medios de pago, debe cumplir con los estándares de seguridad PCI DSS.
* OR-10: Los términos y condiciones de uso deben estar disponibles para todos los usuarios al momento de registrarse.

11.4 Requerimientos de Reúso de Componentes

* OR-11: El sistema debe ser diseñado bajo principios de modularidad, permitiendo la reutilización de componentes (ej. autenticación, notificaciones).
* OR-12: Se recomienda el uso de componentes open-source reconocidos (frameworks, librerías) para acelerar el desarrollo.
* OR-13: Los módulos de reportes y notificaciones deberán diseñarse para ser reutilizados en otros sistemas académicos de DuocUC.

1. **Glosario**

* API (Application Programming Interface): Conjunto de funciones y procedimientos que permiten la comunicación entre diferentes aplicaciones de software.
* App: Aplicación móvil diseñada para ejecutarse en dispositivos como teléfonos inteligentes o tablets.
* BD (Base de Datos): Conjunto de datos organizados y relacionados entre sí, que permiten su almacenamiento y consulta eficiente.
* CRUD (Create, Read, Update, Delete): Operaciones básicas de gestión de datos en un sistema.
* Dashboard: Panel de visualización gráfica que muestra indicadores clave de un sistema (ej. inscripciones, asistencia).
* DuocUC: Institución de educación superior en Chile, contexto de aplicación de este proyecto.
* Estudiante: Usuario final que se inscribe en ayudantías para reforzar asignaturas.
* Helpdesk / Soporte: Función de asistencia técnica para resolver incidencias de usuarios.
* JWT (JSON Web Token): Estándar de seguridad para la autenticación y autorización de usuarios en sistemas distribuidos.
* Ley 19.628: Ley chilena sobre protección de la vida privada y de datos personales.
* Mockup / Wireframe: Representación visual o prototipo de una interfaz de usuario antes de su desarrollo.
* Notificación Push: Mensaje enviado por un servidor a un dispositivo móvil, incluso si la aplicación no está abierta.
* PCI DSS: Estándar internacional de seguridad de datos para la industria de tarjetas de pago.
* Responsividad: Capacidad de un sitio web o app de adaptarse a distintos tamaños de pantalla y dispositivos.
* Scrum: Metodología ágil de gestión de proyectos basada en ciclos iterativos (sprints).
* SQL (Structured Query Language): Lenguaje estándar para la gestión de bases de datos relacionales.
* Supervisor: Usuario encargado de generar consultas y reportes sobre el funcionamiento del sistema (docente o coordinador).
* UI (User Interface): Interfaz de usuario, punto de interacción entre personas y un sistema.
* UX (User Experience): Experiencia de usuario, grado de satisfacción y facilidad en el uso del sistema.