



# Índice de Contenidos

[**Indice de Contenidos 1**](#_gjdgxs)

[**Información del proyecto 2**](#_30j0zll)

[Datos 2](#_1fob9te)

[Patrocinadores 2](#_3znysh7)

[Gerente de Proyecto 2](#_2et92p0)

[Niveles de autoridad 2](#_tyjcwt)

[Lista de Interesados (stakeholders) 2](#_1t3h5sf)

[Cronograma de hitos principales 2](#_2s8eyo1)

[Presupuesto estimado 3](#_3rdcrjn)

[**Descripción del proyecto 3**](#_26in1rg)

[Objetivos del Negocio 3](#_lnxbz9)

[Justificación del proyecto – Contexto 3](#_35nkun2)

[Problema-Necesidad 3](#_1ksv4uv)

[**Descripción del producto 3**](#_44sinio)

[Solución Propuesta 3](#_2jxsxqh)

[Objetivos del proyecto 3](#_z337ya)

[Objetivos de desarrollo 4](#_1y810tw)

[Entregables 4](#_4i7ojhp)

[**Descripción del sistema 4**](#_2xcytpi)

[Requerimientos de alto nivel 4](#_1ci93xb)

[Premisas y restricciones 5](#_3whwml4)

[Riesgos iniciales de alto nivel 5](#_2bn6wsx)

[Especificaciones técnicas de las herramientas de desarrollo 5](#_qsh70q)

[Tipo de Hardware 5](#_3as4poj)

[Tipo de interfaz de Software 5](#_1pxezwc)

[Tipo de interfaz de Usuario 5](#_49x2ik5)

[**Requisitos de aprobación del proyecto 5**](#_2p2csry)

[**Aprobaciones y control de cambios 5**](#_147n2zr)

# Información del proyecto

## Datos

| Archivo:Mitsui Group logo.svg - Wikipedia, la enciclopedia libre | Empresa / Organización | BITNOVA |
| --- | --- | --- |
| Nombre del Proyecto | Monedaventura – Juego de Educación Financiera para Niños |
| Duración | 18 semanas (agosto – diciembre) |
| Metodología de Trabajo | Ágil – Scrum (gestionado mediante Jira Software) |
| Tipo de Proyecto | Proyecto de Título – Ingeniería en Informática |
| Institución | Instituto Profesional / Duoc UC (Sede Padre Alonso de Ovalle) |
| Equipo de Proyecto | Tyhara Mujica, Kevin Ortega, Alberto Lefio |
| Versión del documento | 1.0 |

## Patrocinadores

| **Nombre** | **Cargo** | **Departamento / División** |
| --- | --- | --- |
| Alberto Lefio | Scrum Master / Líder de Proyecto | Alta (coordinación, control de cronograma y cambios). |
| Tyhara Mujica | Diseñadora UI/UX y Desarrolladora Frontend | Media (responsable del diseño visual y la interfaz). |
| Kevin Ortega | QA Tester / Analista de requisitos | Media (control de calidad, pruebas y documentación). |

## 

## Lista de Interesados (stakeholders)

| **Nombre** | **Rol** | **Interés en el proyecto** | **Influencia** | **Estrategia de comunicación** |
| --- | --- | --- | --- | --- |
| Docente Guía | Supervisor | Aprobación del proyecto | Alta | Reuniones semanales de revisión (viernes). |
| Equipo del Proyecto | Desarrolladores | Aprobación del título, experiencia técnica | Alta | Comunicación diaria vía Jira, Discord, Reuniones Presenciales. |
| Usuarios (Niños) | Usuarios finales | Aprendizaje y diversión | Media | Pruebas simuladas de usabilidad. |
| Padres / Tutores | Supervisores | Seguridad del contenido | Media | Revisión de reportes y panel parental. |

## 

## Cronograma de hitos principales

| **Hito** | **Semana** | **Entregable** |
| --- | --- | --- |
| Prototipo validado | 5 | Diseño Figma y flujo del juego |
| MVP funcional | 8 | Primer nivel jugable |
| Versión con múltiples niveles | 14 | Juego educativo ampliado |
| Entrega y presentación final | 18 | Producto completo + documentación |

## 

## Presupuesto estimado

| El proyecto no contempla presupuesto monetario, dado su carácter académico.  Sin embargo, se estiman los siguientes recursos no financieros |
| --- |

## 

| **Recurso** | **Tipo** | **Costo estimado** | **Observación** |
| --- | --- | --- | --- |
| Herramientas de software | Open Source (React, Django, PostgreSQL) | $0 | Licencias libres |
| Diseño UI (Figma) | Gratuito académico | $0 | Uso en entorno educativo |
| Horas de trabajo de los estudiantes | Aporte personal | – | Dedicación de tiempo semanal |

# Descripción del proyecto

## 

| Monedaventura es un juego educativo digital diseñado para enseñar conceptos financieros básicos a niños de entre 8 y 12 años, mediante dinámicas de gamificación.  Combina aprendizaje, entretenimiento y tecnología para fomentar el ahorro, la planificación y el consumo responsable desde edades tempranas. |
| --- |

## Objetivos del Negocio

| La educación financiera en Chile presenta brechas significativas, especialmente en los niveles escolares.  El uso de recursos digitales interactivos ofrece una oportunidad para enseñar estos conceptos de forma atractiva. |
| --- |

## Justificación del proyecto – Contexto

| Contribuir al desarrollo de la educación financiera infantil a través de un recurso digital lúdico e interactivo, que promueva la formación de hábitos económicos positivos y accesibles para la comunidad educativa chilena. |
| --- |

## Problema-Necesidad

| Chile es un país sísmico, que corre el riesgo que se caigan los sistemas afectando a la producción, al ser un país importante para la empresa por la cantidad de recursos que existe en este, es vital mantener una conexión constante. |
| --- |

## 

## Solución Propuesta

| Desarrollar un juego educativo digital multiplataforma, con misiones y recompensas, que permita a los niños comprender la gestión del dinero de manera práctica y divertida. |
| --- |

# Descripción del producto

## 

| Juego educativo con los siguientes componentes:   * Interfaz infantil e intuitiva. * Misiones y mini-juegos basados en decisiones financieras. * Sistema de monedas virtuales y logros. * Panel parental para supervisión y validación. * Evaluaciones de aprendizaje antes y después de cada nivel. |
| --- |

## Objetivos del proyecto

| **Objetivo** | **Indicador de éxito** |
| --- | --- |
| **Alcance** | |
| Entregar un juego funcional con 3–5 niveles educativos. | 100% de los requerimientos funcionales implementados. |

# 

| **Objetivo** | **Indicador de éxito** |
| --- | --- |
| **Calidad** | |
| Construir Software: Asegurar la correcta estructura técnica del sistema, aplicando buenas prácticas de codificación, modularidad y control de versiones. | Código validado sin errores críticos; repositorio organizado con >90% de cumplimiento en revisiones técnicas. |
| Desarrollar Software: Garantizar la funcionalidad completa y el rendimiento óptimo del producto final, priorizando la experiencia del usuario. | Sistema estable con carga <3 segundos; ≥80% éxito en pruebas de usabilidad y QA. |

| **Objetivo** | **Indicador de éxito** |
| --- | --- |
| **Cronograma** | |
| Construcción de Software: Ejecutar la construcción del software según el plan establecido. | Estructura finalizada conforme a la arquitectura definida. |
| Desarrollo del software: Implementar y desplegar el sistema educativo funcional. | Software en línea, completamente operativo y validado. |
| Seguridad del software: Proteger los datos de los usuarios infantiles cumpliendo normativas COPPA y GDPR. | Aprobador de test de calidad y verificación de estándares de seguridad. |
| **Tiempos de Desarrollo** |  |
| Cumplir los plazos establecidos en el cronograma general del proyecto. | Plazos cumplidos según la carta Gantt (sin retrasos significativos). |

| **Objetivo** | **Indicador de éxito** | |
| --- | --- | --- |
| **Costos** | | |
| Minimizar los gastos mediante herramientas gratuitas y recursos propios. | | 0% de costos externos o licencias comerciales. |

## Objetivos de desarrollo

| Aplicar metodologías ágiles y herramientas profesionales en la gestión y ejecución del proyecto. |
| --- |

## Entregables

| Entregable | Descripción |
| --- | --- |
| Centro de servidores terminado | Infraestructura de servidores (o configuración en la nube) con la arquitectura requerida para el despliegue, backup y sincronización entre nodos. Incluye documentación de configuración y credenciales administrativas (entregadas al patrocinador/institución). |
| Software terminado | Versión final del juego desplegada y operativa (PWA / hosting), con base de datos sincronizada, panel parental, módulos de misiones, sistema de monedas y pruebas de aceptación aprobadas. |

# Descripción del sistema

## Requerimientos de alto nivel

| Resumen funcional: El sistema Monedaventura es una aplicación web progresiva (PWA) cliente-servidor que permite a niños (8–12 años) registrarse (con verificación parental), realizar misiones y mini-juegos, ganar monedas virtuales, y registrar progresos. Los padres acceden a un panel para supervisar y configurar límites. El backend gestiona usuarios, sesiones, sincronización offline/online, analíticas anónimas y evaluación de aprendizaje.  Arquitectura técnica principal:   * Frontend: React / Next.js (PWA), responsive para móviles y tablets. * Backend: Django REST Framework (APIs REST). * Base de datos: PostgreSQL con mecanismos de sincronización e historial. * Despliegue: Servidores distribuidos (o cloud) con balanceo y replicación. |
| --- |

| **ID** | **Requisito** | **Definición** | **Prioridad** | **Relacionado con** |
| --- | --- | --- | --- | --- |
| RF01 | Registro y Perfil de Usuario | Definición: Permite al niño crear un perfil con nombre, edad, avatar y PIN simple; requiere validación parental vía email. Incluye onboarding con tutorial animado. Por qué: Asegura seguridad y personalización inicial. Cómo se relaciona: Cumple COPPA/GDPR; base para progreso guardado. | Alta | UC-01; Interfaz intuitiva (OS1) |
| RF02 | Gestión de Misiones y Mini-Juegos | Definición: Presenta misiones diarias/semanas con mini-juegos (ej. "Mercadito" para presupuesto). Cada uno enseña un concepto (ahorro, necesidades vs. deseos). Por qué: Core loop gamificado para aprendizaje activo. Cómo se relaciona: Refuerza decisiones financieras; evita loot boxes. | Alta | UC-02; Niveles progresivos (OS3) |
| RF03 | Sistema de Monedas Virtuales y Ahorro | Definición: Niños ganan/gastan monedas virtuales; divide en categorías (ahorro, gasto, metas). Barra de progreso visual. Por qué: Simula finanzas reales de forma lúdica. Cómo se relaciona: Integra con logros; no convertible a dinero real. | Alta | UC-03; Sistema de misiones (OS2) |
| RF04 | Sistema de Logros y Recompensas | Definición: Otorga badges/logros por completitud (ej. "Planificador Junior"); visualiza en barra/gráficos. Por qué: Motiva intrínsecamente (SDT: competencia). Cómo se relaciona: Recompensas predecibles, no aleatorias. | Alta | UC-02; Refuerzo de decisiones |
| RF05 | Panel Parental | Definición: Dashboard para padres: ver progreso, validar recompensas, configurar límites de tiempo. Por qué: Fomenta transferencia a vida real y supervisión ética. Cómo se relaciona: Notificaciones y reportes (ej. PDF exportable). | Media | UC-04; Validación usabilidad (OS4) |
| RF06 | Analytics y Registro de Eventos | Definición: Registra eventos (sesiones, misiones completadas, ahorro vs. gasto) para métricas internas. Por qué: Permite iteraciones basadas en datos. Cómo se relaciona: Anonimizado; soporta evaluación pre/post. | Media | UC-05; Documentación (OS5) |
| RF07 | Evaluación de Aprendizaje | Definición: Administra tests cortos (5-8 preguntas) pre/post por nivel; calcula puntaje y notifica padres. Por qué: Mide impacto (mejora +30%). Cómo se relaciona: Integra con niveles; usa preguntas múltiples. | Media | UC-05; Métricas de impacto |
| RF08 | Modo Multijugador Opcional | Definición: Comparar progresos anónimos con amigos (sin datos personales). Por qué: Fomenta relación (SDT). Cómo se relaciona: Baja prioridad; no en MVP inicial. | Baja | Innovación social |
| RF09 | Guardado Automático y Offline | Definición: Guarda progreso en BD; modo offline para misiones básicas con sincronización. Por qué: Mejora accesibilidad. Cómo se relaciona: Usa PostgreSQL; reanuda sesiones. | Alta | Factibilidad técnica |
| RF10 | Integración Educativa | Definición: Enlaces opcionales para padres a fuentes (ej. OECD/INFE). Por qué: Valida contenido. Cómo se relaciona: Alineado con revisión bibliográfica. | Baja | Fundamentación |

## Premisas y restricciones

| Premisas   * El equipo dispone de acceso a internet y computadores personales para desarrollo. * El docente guía validará entregables y hitos los viernes. * Se emplearán herramientas gratuitas o con licencia académica (Figma, GitHub, Jira).   Restricciones   * Base de datos interconectada con distintos servidores del mundo. El sistema deberá considerar sincronización y replicación de datos entre diferentes nodos/servicios (por ejemplo, replicación geográfica o uso de CDN). * No se cuenta con presupuesto para infraestructura física dedicada (si se requiere, usar soluciones en la nube gratuitas/educativas o servidores compartidos). * Alcance limitado a la entrega académica; no incluye soporte comercial post-entrega. |
| --- |

## Riesgos iniciales de alto nivel

| ID | Riesgo | Probabilidad | Impacto | Estrategia / Plan de Mitigación |
| --- | --- | --- | --- | --- |
| R-01 | Desarrollar un software no compatible con el hardware o servidores de otros países. | Media-Alta | Alta | Realizar pruebas en entornos virtuales y en servidores de diferentes configuraciones. Documentar requerimientos mínimos de compatibilidad. |
| R-02 | Retraso en la entrega de tareas debido a carga académica u otros compromisos. | Alta | Alta | Planificación semanal en Jira, uso de sprints cortos y revisiones los viernes. Redistribución de tareas si hay atrasos. |
| R-03 | Pérdida de datos por fallas en la base de datos o sincronización. | Media | Alta | Configurar respaldos automáticos, replicación entre servidores y pruebas de restauración. |
| R-04 | Fallos en la sincronización de la base de datos interconectada globalmente. | Media | Alta | Implementar políticas de consistencia eventual, priorizar sincronización incremental y validación post-envío. |
| R-05 | Incompatibilidad o errores entre las versiones de las herramientas (React, Django, PostgreSQL). | Media | Media | Mantener control de versiones estable, usar entornos virtuales (venv), y pruebas de integración antes del despliegue. |
| R-06 | Falta de cumplimiento con normativas de privacidad infantil (COPPA/GDPR). | Media | Alta | Minimizar recolección de datos, incluir consentimiento parental, y revisar requerimientos legales antes de la entrega. |
| R-07 | Problemas de seguridad o vulnerabilidad en el sistema. | Media | Alta | Aplicar cifrado en base de datos y comunicaciones (SSL/TLS), realizar test de seguridad y auditoría básica. |
| R-08 | Escasa disponibilidad o fallas del servidor compartido o cloud durante pruebas. | Media | Media | Contar con backup local y espejo del repositorio, usar entornos alternativos (localhost, contenedores). |
| R-09 | Baja participación o disponibilidad del docente guía o revisores del proyecto. | Baja | Media | Mantener comunicación constante vía correo institucional y Jira; registrar entregas con fechas claras. |
| R-10 | Dificultades de integración entre diseño (Figma) y desarrollo final. | Media | Media | Revisión UI/UX semanal, prototipado validado antes de codificar, colaboración entre diseñador y desarrolladores. |

## Especificaciones técnicas de las herramientas de desarrollo

| * Procesador: 2ª generación de procesadores Intel® Xeon® Scalable, Single o Dual Socket. * Memoria RAM: hasta 2 TB DDR4 2933MHz, quad channel, ECC (según la carga prevista). * GPU (opcional): Soporta hasta 4 GPU NVIDIA GeForce o NVIDIA Quadro (8 según chasis) — solo si se requiere procesamiento de gráficos o ML. * Chipset: Intel® C612 / C621. * Conectividad de red: Single o dual 1 Gbps / 10 Gbps Ethernet (según SLA). * Almacenamiento: SATA3 ports con soporte RAID 0/1/5/10; preferible SSD NVMe para bases de datos. * Puertos: USB 2.0 & USB 3.0. * Redundancia: RAID + backups automáticos y snapshots.   Servidor para base de datos y sincronización   * CPU: 8–16 vCPU (según carga). * RAM: 32–128 GB (según volumen de usuarios simulados). * Almacenamiento: SSD NVMe, IOPS optimizados; tamaño acorde al volumen de datos. * Conectividad: 1–10 Gbps, latencia baja. |
| --- |

## Tipo de Interfaz de Hardware

| * Servidor compartido (ejemplo) * CPU: Intel® Xeon® Scalable (multi-core). * Memoria: 2 TB RAM DDR4 2933MHz (según la configuración del proveedor). * Almacenamiento: SSD 32 GB (mínimo para servidor de pruebas; mayor para producción). * Red: 1 Gbps mínimo.   Dispositivos del usuario final   * Smartphones / Tablets: Android (8+) / iOS (12+) / navegadores modernos (Chrome, Safari, Firefox). * PC: Windows 10+, macOS recientes. |
| --- |

## Tipo de Interfaz de Software

| Servidor compartido   * Sistema operativo: CentOS 7.6 (o versión estable LTS similar) — preferible usar versiones soporte o imágenes en Debian/Ubuntu LTS según proveedor. * Stack web: Apache / LAMP o Nginx + Gunicorn/ASGI para Django. * Panel de control (opcional): WHM / cPanel para hosting compartido (no obligatorio si se usa cloud). * Base de datos: PostgreSQL v13+ (configurada para replicación y backups). * Servicios: Redis (cache / colas), Nginx/Traefik (proxy), Certbot para TLS.   Navegadores / clientes (usuarios finales):   * Internet Explorer 10+ (se sugiere no priorizar IE, mejor compatibilidad con Edge/Chrome/Firefox/Safari) * Firefox (últimas versiones) * Google Chrome (últimas versiones) * Safari (iOS/macOS) |
| --- |

## Tipo de Interfaz de Usuario

| * Buscador amigable para localizar contenidos, misiones o artículos del panel educativo. * Interfaz infantil: botones grandes (>44px), tipografía legible, contraste adecuado, audio de apoyo. * Controles accesibles: modo lectura, soporte para daltonismo, subtítulos en audios. * Tutorial interactivo (onboarding): animado, con pasos guiados y PIN parental para registrar cuenta. * Panel parental: dashboards con gráficos simples (progreso, monedas ahorradas, sesiones). |
| --- |

# Requisitos de aprobación del proyecto

El proyecto requiere las siguientes aprobaciones antes de su entrega final:

| **Aprobación** | **Responsable** | **Criterio** |
| --- | --- | --- |
| Aprobación del plan y cronograma | Docente Guía | Cronograma validado y plan de trabajo aprobado. |
| Aprobación del entorno de servidores / data center | Equipo de proyecto | Configuración de servidores verificada |
| Aprobación de pruebas de calidad | Equipo de proyecto / Docente | Resultados de pruebas funcionales y de usabilidad aprobados. |
| Aprobación final del proyecto (defensa) | Comisión evaluadora | Documento final y demo aceptados. |

# Aprobaciones y control de cambios

| **Versión** | **Nombre** | **Rol** | **Fecha** | **Firma** |
| --- | --- | --- | --- | --- |
| 1.0 | Docente Guía | Patrocinador académico | [dd-mm-aaaa] | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 1.1 | Alberto Lefio | Scrum Master / Líder de Proyecto | [dd-mm-aaaa] | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 1.2 | Tyhara Mujica | Desarrolladora UI/UX | [dd-mm-aaaa] | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 1.3 | Kevin Ortega | QA / Analista | [dd-mm-aaaa] | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |