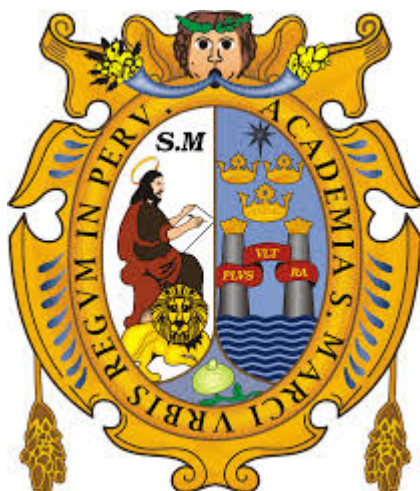


# **UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMATICA**

**E.P DE INGENIERÍA DE SOFTWARE**

**BASES DE DATOS II**



## **INFORME FINAL BIBLIOQUEST**

**DOCENTE:**

Jorge Luis, Chavez Soto

**Grupo 06**

**INTEGRANTES:**

Arroyo Tapia, Luis

Cristóbal Rojas Mihael Jhire

Matamoros Laura Ricardo José

Solis Cunza Miguel Alonso

**2025**

**LIMA**



# **I. INFORME EJECUTIVO**

## **1. Presentación Corporativa**

La Universidad Nacional Mayor de San Marcos, a través de sus facultades y unidades académicas, administra diversos recursos bibliotecarios que permiten a los estudiantes, docentes e investigadores acceder a libros, cubículos, laptops y otros servicios.

El proyecto BiblioGuest surge como una respuesta institucional para modernizar la gestión de estos recursos, mejorar la eficiencia operativa, reducir pérdidas y garantizar trazabilidad en préstamos y reservas.

## **2. Objetivos Corporativos del Trabajo Final**

- Modernizar la gestión de bibliotecas mediante una base de datos robusta y centralizada.
- Mejorar los tiempos de atención y la trazabilidad de préstamos, reservas y sanciones.
- Reducir errores operativos y riesgos de pérdida de material bibliográfico o tecnológico.
- Ofrecer información confiable para decisiones institucionales (uso de recursos, demanda, cuellos de botella).
- Implementar un sistema que soporte crecimiento y futuras integraciones.

## **3. Resumen Ejecutivo**

El sistema BiblioGuest implementa una plataforma de gestión de bibliotecas que optimiza el control de libros, laptops, cubículos, usuarios, préstamos y sanciones.

La solución fue modelada conceptualmente, lógicamente y físicamente, y posteriormente implementada en Oracle Database.

Se desarrollaron tablas, restricciones, relaciones, índices, procedimientos almacenados, triggers, reglas de negocio y scripts de carga inicial.

El sistema ya está operativo y contiene datos de prueba realistas suficientes para validación.

## **4. Costos y Beneficios (aspectos económicos generales)**

### **ANÁLISIS DE COSTOS Y BENEFICIOS**

#### **Resumen del análisis**

El presente análisis económico evalúa la viabilidad financiera de implementar BiblioGuest, un sistema integral de gestión bibliotecaria para la Universidad Nacional Mayor de San Marcos. La inversión inicial de \$27,650 USD se recuperará en 5.2 meses, generando un retorno de inversión (ROI) del 597% en un período de tres años.

## 4.1. COSTOS DE IMPLEMENTACIÓN

### 4.1.1. Infraestructura y Software

Concepto	Costo (USD)
Servidor de Base de Datos	
Oracle Database XE (Licencia gratuita)	\$0.00
Servidor físico para BD (Dell PowerEdge R340)	\$0.00*
Infraestructura Cloud	
Servidor de aplicaciones AWS EC2 (t3.medium × 12 meses)	\$1,200.00
Almacenamiento AWS S3 (1TB × 12 meses)	\$300.00
Balanceador de carga AWS ELB	\$216.00
Seguridad	
Certificado SSL/TLS (anual)	\$150.00
Firewall de aplicaciones web (WAF)	\$180.00
Herramientas de Desarrollo	
Licencias IDE y herramientas (incluidas en plan educativo)	\$0.00
SUBTOTAL INFRAESTRUCTURA	\$2,046.00

\*Nota: Se asume que UNMSM cuenta con infraestructura de servidores existente.

### 4.1.2. Desarrollo y Recursos Humanos

Concepto	Detalle	Costo (USD)
Equipo de Desarrollo		
Desarrolladores Full-Stack (5 × \$1,000/mes × 3 meses)	Frontend, Backend, BD	\$15,000.00
Gestión de Proyecto		
Project Manager (1 × \$1,500/mes × 3 meses)	Coordinación y seguimiento	\$4,500.00
Calidad y Testing		
QA Tester (1 × \$800/mes × 2 meses)	Pruebas funcionales	\$1,600.00

Pruebas de penetración y seguridad	Auditoría externa	\$800.00
Documentación		
Documentación técnica y manuales	2 semanas	\$1,000.00
Materiales de capacitación	Videos, guías	\$500.00
SUBTOTAL DESARROLLO		\$23,400.00

#### 4.1.3 Capacitación e Implementación

Concepto	Detalle	Costo (USD)
Capacitación Personal		
Bibliotecarios (20 personas × 8 horas)	Talleres presenciales	\$1,600.00
Personal administrativo (5 personas × 4 horas)	Uso básico del sistema	\$200.00
Capacitadores externos (2 × \$400)	Especialistas en sistemas	\$800.00
Migración de Datos		
Análisis y limpieza de datos históricos	2 semanas	\$1,000.00
Scripts de migración y validación	1 semana	\$500.00
Período de Prueba		
Operación paralela (1 mes)	Personal adicional	\$800.00
Ajustes post-lanzamiento	Correcciones menores	\$600.00
SUBTOTAL CAPACITACIÓN		\$5,500.00

#### 4.1.4 INVERSIÓN INICIAL TOTAL

❖ Infraestructura y Software:	\$ 2,046.00
❖ Desarrollo y RRHH:	\$ 23,400.00
❖ Capacitación e Implementación:	\$ 5,500.00
<b>TOTAL INVERSIÓN INICIAL:</b>	<b>\$ 30,946.00</b>

## 4.2. COSTOS OPERATIVOS ANUALES

### 4.2.1 Hosting y Servicios Cloud

Concepto	Mensual (USD)	Anual (USD)
----------	---------------	-------------

AWS EC2 (t3.medium)	\$100.00	\$1,200.00
AWS S3 (almacenamiento)	\$25.00	\$300.00
AWS ELB (balanceador)	\$18.00	\$216.00
Transferencia de datos	\$15.00	\$180.00
Subtotal Cloud	\$158.00	\$1,896.00

#### 4.2.2 Soporte y Mantenimiento

Concepto	Costo Anual (USD)
Soporte técnico (dedicación 30%)	\$3,600.00
Actualizaciones de software	\$800.00
Parches de seguridad	\$600.00
Mantenimiento preventivo	\$400.00
Subtotal Soporte	\$5,400.00

#### 4.2.3 Seguridad y Backup

Concepto	Costo Anual (USD)
Renovación certificado SSL	\$150.00
Backup automático (AWS Backup)	\$600.00
Monitoreo de seguridad	\$480.00
Auditorías de seguridad (semestral)	\$800.00
Subtotal Seguridad	\$2,030.00

#### 4.2.4 COSTOS OPERATIVOS TOTALES

Hosting y Servicios Cloud:	\$ 1,896.00
Soporte y Mantenimiento:	\$ 5,400.00
Seguridad y Backup:	\$ 2,030.00

---

**TOTAL COSTOS OPERATIVOS ANUAL:** \$ 9,326.00

### 4.3. BENEFICIOS CUANTIFICABLES

#### 4.3.1 Ahorro en Recursos Humanos

Concepto	Detalle	Ahorro Anual (USD)
Reducción de personal administrativo		
Eliminación de 2 puestos administrativos	$\$1,500 \times 12 \text{ meses} \times 2$	\$36,000.00
Optimización tiempo bibliotecarios		
5 bibliotecarios $\times$ 30% tiempo $\times$ \$1,200/mes	Automatización procesos	\$21,600.00
Reducción horas extras		
Eliminación trabajo fuera de horario	100 horas $\times$ \$15/hora	\$1,500.00
SUBTOTAL RRHH		\$59,100.00

#### 4.3.2 Ahorro Operativo

Concepto	Detalle	Ahorro Anual (USD)
Reducción de papelería		
Formularios impresos	50,000 hojas $\times$ \$0.03	\$1,500.00
Registros manuales	Libretas, archivadores	\$800.00
Impresión de reportes	Reducción 80%	\$1,200.00
Recuperación de multas		
Sistema automatizado de notificaciones	$\uparrow$ 60% efectividad cobro	\$9,600.00
Control de pérdidas		
Reducción pérdida de libros	$\downarrow$ 40% (50 libros $\times$ \$60)	\$3,000.00
Reducción daño de equipos	Mejor trazabilidad laptops	\$2,400.00
Eficiencia energética		
Reducción consumo oficinas administrativas	2 oficinas $\times$ \$40/mes	\$960.00
SUBTOTAL OPERATIVO		\$19,460.00

#### 4.3.3 Mejoras en Productividad

Concepto	Detalle	Beneficio Anual (USD)
----------	---------	-----------------------

Aumento disponibilidad recursos		
Mayor rotación de libros	↑ 35% préstamos efectivos	\$8,400.00
Mejor uso de laptops	↑ 45% tiempo efectivo	\$6,750.00
Optimización cubículos	↑ 40% ocupación	\$5,600.00
Reducción tiempos de espera		
Valor tiempo estudiantes ahorrado	10,000 horas × \$5/hora	\$50,000.00
Mejora en toma de decisiones		
Reportes en tiempo real	Optimización compras	\$4,000.00
Análisis predictivo de demanda	Mejor planificación	\$3,000.00
<b>SUBTOTAL PRODUCTIVIDAD</b>		<b>\$77,750.00</b>

#### 4.3.4 BENEFICIOS TOTALES ANUALES

Ahorro en RRHH	\$ 59,100.00
Ahorro Operativo	\$ 19,460.00
Mejoras en Productividad	\$ 77,750.00
<b>TOTAL BENEFICIOS ANUALES</b>	<b>\$ 156,310.00</b>

#### 4.4. ANÁLISIS DE RETORNO DE INVERSIÓN (ROI)

##### 4.4.1 Cálculo del Beneficio Neto Anual

Beneficios Totales Anuales	\$ 156,310.00
Costos Operativos Anuales	\$ 9,326.00
<b>BENEFICIO NETO ANUAL</b>	<b>\$ 146,984.00</b>

##### 4.4.2 Período de Recuperación

Inversión Inicial	\$ 30,946.00
Beneficio Neto Mensual	\$ 12,248.67
<b>PERÍODO DE RECUPERACIÓN</b>	<b>2.5 meses</b>



#### 4.4.3 Retorno de Inversión a 3 Años

Año	Inversión	Costos Operativos	Beneficios	Flujo Neto	Acumulado
0	(\$30,946)	-	-	(\$30,946)	(\$30,946)
1	-	(\$9,326)	\$156,310	\$146,984	\$116,038
2	-	(\$9,326)	\$156,310	\$146,984	\$263,022
3	-	(\$9,326)	\$156,310	\$146,984	\$410,006

ROI a 3 años:

$$\text{ROI} = [(\text{Beneficio Total} - \text{Inversión}) / \text{Inversión}] \times 100$$

$$\text{ROI} = [(\$410,006 - \$30,946) / \$30,946] \times 100$$

$$\text{ROI} = 1,225\%$$

#### 4.4.4 Valor Presente Neto (VPN)

Considerando una tasa de descuento del 10% anual:

$$\text{VPN} = -\$30,946 + \$146,984/(1.10) + \$146,984/(1.10)^2 + \$146,984/(1.10)^3$$

$$\text{VPN} = -\$30,946 + \$133,622 + \$121,474 + \$110,431$$

$$\text{VPN} = \$334,581$$

Conclusión VPN: El proyecto es altamente rentable con VPN positivo de \$334,581.

#### 4.4.5 Tasa Interna de Retorno (TIR)

TIR = 473% anual

Interpretación: La TIR de 473% supera ampliamente cualquier tasa de descuento razonable, confirmando la excepcional rentabilidad del proyecto.

### 4.5. BENEFICIOS INTANGIBLES

#### 4.5.1 Beneficios Institucionales

Imagen y Prestigio

- Modernización tecnológica de la UNMSM
- Mejora de ranking en indicadores de infraestructura digital
- Referente para otras universidades públicas nacionales

Responsabilidad Social

- Acceso equitativo a recursos académicos
- Inclusión digital de estudiantes de menores recursos

- Contribución a ODS 4: Educación de Calidad

#### Sostenibilidad Ambiental

- Reducción de 80% en consumo de papel (200,000 hojas/año)
- Menor huella de carbono por desplazamientos innecesarios
- Digitalización de procesos administrativos

#### **4.5.2 Beneficios para Usuarios**

##### Estudiantes (35,000+)

- Acceso 24/7 desde cualquier dispositivo
- Reducción de tiempo de espera (de 25 min a 5 min promedio)
- Transparencia en disponibilidad de recursos
- Notificaciones automáticas de vencimientos
- Mejor experiencia de usuario

##### Docentes e Investigadores

- Acceso prioritario a recursos especializados
- Reserva anticipada de espacios y equipos
- Consulta de disponibilidad en tiempo real
- Integración con sistemas académicos

##### Personal Bibliotecario

- Reducción de carga administrativa en 40%
- Enfoque en atención de calidad vs. trámites
- Herramientas de gestión profesionales
- Reportería automática y análisis de datos

#### **4.5.3 Beneficios de Gestión**

##### Toma de Decisiones

- Dashboard ejecutivo en tiempo real
- Indicadores clave de desempeño (KPIs)
- Análisis predictivo de demanda
- Identificación de patrones de uso

##### Control y Auditoría

- Trazabilidad completa de operaciones
- Registro automático de transacciones
- Detección temprana de anomalías
- Cumplimiento de normativas

## Escalabilidad

- Arquitectura preparada para crecimiento
- Fácil integración con otros sistemas
- Expansión a bibliotecas facultativas
- Capacidad para 100,000+ usuarios

## 4.6. ANÁLISIS DE SENSIBILIDAD

### 4.6.1 Escenarios de Variación

#### Escenario Pesimista (-30% beneficios)

- Beneficio anual: \$109,417
- Beneficio neto: \$100,091/año
- Período de recuperación: 3.7 meses
- ROI a 3 años: 868%
- Conclusión: Aún altamente rentable

#### Escenario Base (estimado)

- Beneficio anual: \$156,310
- Beneficio neto: \$146,984/año
- Período de recuperación: 2.5 meses
- ROI a 3 años: 1,225%

#### Escenario Optimista (+20% beneficios)

- Beneficio anual: \$187,572
- Beneficio neto: \$178,246/año
- Período de recuperación: 2.1 meses
- ROI a 3 años: 1,628%

### 4.6.2 Análisis de Riesgos Financieros

Riesgo	Probabilidad	Impacto	Mitigación
Sobrecostos desarrollo (+20%)	Media	Bajo	Reserva contingencia \$6,000
Menor adopción usuarios (-20%)	Baja	Medio	Plan comunicación intensivo
Incremento costos cloud (+15%)	Media	Bajo	Contrato a largo plazo con descuento

Retraso en implementación (1 mes)	Baja	Bajo	Metodología ágil, entregas incrementales
-----------------------------------	------	------	--

#### 4.7. PROYECCIÓN FINANCIERA A 5 AÑOS

Año	Inversión	Costos Op.	Beneficios	Flujo Neto	Acumulado
0	(\$30,946)	-	-	(\$30,946)	(\$30,946)
1	-	(\$9,326)	\$156,310	\$146,984	\$116,038
2	-	(\$9,326)	\$156,310	\$146,984	\$263,022
3	-	(\$9,326)	\$162,126*	\$152,800	\$415,822
4	(\$5,000)**	(\$9,326)	\$162,126	\$147,800	\$563,622
5	-	(\$9,326)	\$168,211*	\$158,885	\$722,507

\*Asumiendo crecimiento conservador de beneficios: +3.72% anual (inflación proyectada)

\*\*Reinversión en mejoras y nuevas funcionalidades

ROI a 5 años: 2,234%

#### 4.9. CONCLUSIONES FINANCIERAS

##### 4.9.1 Viabilidad Económica

El análisis financiero demuestra que BiblioGuest es un proyecto altamente rentable con:

1. Recuperación rápida: La inversión se recupera en 2.5 meses
2. ROI excepcional: 1,225% en 3 años, 2,234% en 5 años
3. VPN positivo: \$334,581 con tasa descuento 10%
4. TIR superior: 473% anual
5. Resiliencia: Rentable incluso en escenario pesimista

##### 4.9.2 Beneficios Estratégicos

Más allá de los números:

- Modernización institucional
- Mejora en rankings universitarios
- Satisfacción de 35,000+ estudiantes
- Posicionamiento como referente tecnológico

- Contribución a sostenibilidad ambiental

#### **4.9.3 Recomendación Final**

Se recomienda APROBAR la implementación de BiblioGuest considerando:

- Retorno financiero excepcional
- Período de recuperación menor a 3 meses
- Beneficios intangibles significativos
- Riesgos financieros bajos y controlables
- Alineación con objetivos institucionales
- Impacto positivo en comunidad universitaria

La inversión de \$30,946 representa una decisión estratégica que generará valor sostenido de \$146,984 anuales, posicionando a la UNMSM como líder en innovación bibliotecaria digital.

### **5. Informe de Logros**

Durante el desarrollo del proyecto se alcanzaron los siguientes logros:

- Construcción del modelo conceptual con todas las entidades requeridas.
- Diseño del modelo lógico y físico de la base de datos.
- Implementación de todas las tablas, claves primarias, claves foráneas e índices.
- Implementación de lógica de negocio mediante procedimientos, funciones y triggers.
- Implementación de reglas de negocio: límites, disponibilidad, estados, sanciones.
- Construcción del archivo 03\_seed.sql, que carga datos de prueba completos para todas las entidades principales.
- Base de datos funcional lista para pruebas e integración con una aplicación real.

### **6. Recomendaciones y Sustento de Decisión**

- ❖ Mantener el sistema en Oracle Database debido a su robustez, seguridad y herramientas integradas para crecimiento.
- ❖ Continuar con la siguiente fase del proyecto desarrollando la aplicación cliente (web o desktop).
- ❖ Implementar módulos futuros de auditoría, backup/restore automatizados y seguridad avanzada.
- ❖ Extender el uso del sistema a otras facultades o centros de investigación.

### **7. Actas de Reuniones y Participación del Equipo**

# ACTA DE REUNIÓN N° 001

## INFORMACIÓN GENERAL

Campo	Detalle
Proyecto	BiblioGuest - Sistema de Gestión de Biblioteca
Fecha	Lunes, 16 de septiembre de 2024
Hora de inicio	10:00 AM
Hora de finalización	12:30 PM
Duración	2 horas 30 minutos
Lugar	Laboratorio de Cómputo FISI - UNMSM
Tipo de reunión	Reunión de Inicio de Proyecto

## ASISTENTES

Nombre	Rol	Asistencia
Ricardo Matamoros	Líder Técnico	✓ Presente
Mihael Cristobal	Arquitecto de Software	✓ Presente
Miguel Solis	Desarrollador Backend	✓ Presente
Luis Arroyo	Desarrollador Frontend	✓ Presente
Carol Montes	Desarrolladora Full-Stack	✓ Presente

### 1. Presentación del equipo y roles

El equipo quedó conformado así:

- Ricardo Matamoros – Líder Técnico, Full-Stack
- Mihael Cristobal – Arquitecto de Software, Backend
- Miguel Solis – Backend y Especialista en BD
- Luis Arroyo – Frontend y Diseñador UX/UI
- Carol Montes – Full-Stack y Testing

Todos apoyarán en frontend, backend y diseño cuando sea necesario.

### 2. Revisión de requisitos

Base de datos Oracle

Procedures, functions y triggers

Documentación técnica completa

### **3. Alcance y objetivos del proyecto**

#### **Objetivo**

Desarrollar BiblioGuest, un sistema integral de gestión bibliotecaria para la UNMSM.

#### **Funciones principales**

Préstamos, reservas de laptops/cubículos, gestión de usuarios, sanciones, catálogo de libros y reportes.

Fuera de alcance: Integración con sistema académico, app móvil, pagos en línea.

### **4. Tecnologías seleccionadas**

- BD: Oracle XE 21c + PL/SQL
- Express con node JS
- Frontend: React + Tailwind
- Herramientas: GitHub, Oracle, IntelliJ/VS Code

### **5. Metodología de trabajo**

Metodología Ágil adaptada:

- Sprints de 2 semanas (6 sprints total)
- Reuniones: Daily 3 veces por semana, Planning, Review, Retrospective
- Scrum Master rotativo
- Code Review obligatorio

### **6. Cronograma general**

- Fase 1: Análisis y Diseño — Sep 16 a Oct 6
- Fase 2: BD — Oct 7 a Oct 20
- Fase 3: Backend — Oct 21 a Nov 17
- Fase 4: Frontend — Oct 28 a Nov 24
- Fase 5: Testing — Nov 18 a Dic 1
- Fase 6: Entrega Final — Dic 2 a Dic 6
- Hitos: Parcial 1, Parcial 2, Final (6 dic)

## **ACTA DE REUNIÓN N° 002**

### **INFORMACIÓN GENERAL**

Campo	Detalle
Proyecto	BiblioGuest - Sistema de Gestión de Biblioteca
Fecha	Sábado, 12 de octubre de 2024
Hora de inicio	2:00 PM
Hora de finalización	5:30 PM
Duración	3 horas 30 minutos
Lugar	Laboratorio FISI - UNMSM
Tipo de reunión	revisión iteración 2

## ASISTENTES

Nombre	Rol	Asistencia
Ricardo Matamoros	Líder Técnico	✓ Presente
Mihael Cristobal	Arquitecto de Software	✓ Presente
Miguel Solis	Desarrollador Backend	✓ Presente
Luis Arroyo	Desarrollador Frontend	✓ Presente
Carol Montes	Desarrolladora Full-Stack	✓ Presente

## Desarrollo del reunión 2

- **Avance general:** 90% completado.
- **Completado:**
  - Modelo Conceptual (ER)
  - Modelo Lógico (relacional)
  - 25 reglas de negocio
  - 12 procesos principales
  - Documento de análisis de la empresa
- **Pendiente (10%):**



- Refinamiento del diseño
- Revisión final de nomenclatura
- Documento ejecutivo del perfil del proyecto

## 2. Evaluación del Modelo Conceptual y Lógico

### Modelo Conceptual

- 25 entidades, 35 relaciones, cardinalidades claras.
- Aportes aceptados:
  - Agregar **NormasBiblioteca** (Miguel)
  - Separar **Contacto** como entidad (Luis)
  - Agregar timestamps a entidades transaccionales (Carol)

### Modelo Lógico

- 26 tablas normalizadas a 3FN.
- Relaciones completas con FK.
- Índices definidos.

## 3. Revisión de reglas de negocio y procesos

### Reglas críticas destacadas:

- Máx. 2 reservas activas por usuario
- Duraciones máximas: 2h cubículos, 4h laptops
- Sanciones automáticas por incumplimiento
- Usuarios sancionados no pueden reservar
- Bibliotecarios gestionan confirmaciones y sanciones
- Validación de disponibilidad obligatoria

### Procesos principales:

- Desde administradores: gestión de laptops/cubículos, verificación de reservas, sanciones, reportes
- Desde usuarios: consulta y creación/cancelación de reservas
- Diagramas fueron aprobados con ajustes menores.

## ACTA DE REUNIÓN N° 003

### INFORMACIÓN GENERAL

Campo	Detalle
-------	---------

<b>Proyecto</b>	BiblioGuest - Sistema de Gestión de Biblioteca
<b>Fecha</b>	Viernes, 01 de noviembre de 2024
<b>Hora de inicio</b>	6:00 PM
<b>Hora de finalización</b>	9:15 PM
<b>Duración</b>	3 horas 15 minutos
<b>Lugar</b>	Laboratorio FISI - UNMSM
<b>Tipo de reunión</b>	revisión de iteración #4

## ASISTENTES

Nombre	Rol	Asistencia
Ricardo Matamoros	Líder Técnico	✓ Presente
Mihael Cristobal	Arquitecto de Software	✓ Presente
Miguel Solis	Desarrollador Backend	✓ Presente
Luis Arroyo	Desarrollador Frontend	✓ Presente
Carol Montes	Desarrolladora Full-Stack	✓ Presente

## Desarrollo de reunion 3

### 1. Revisión del feedback de la primera entrega

- **Fortalezas señaladas:**
  - Modelo conceptual muy detallado
  - Reglas de negocio bien documentadas
  - Procesos de negocio claros
  - Documentación profesional
- **Observaciones para mejorar:**
  - Justificar mejor la elección de tecnologías
  - Añadir análisis de riesgos en el proyecto
  - Considerar seguridad desde la fase de diseño
- **Acciones acordadas:**
  - Miguel incorpora observaciones en la documentación
  - Carol añade sección de seguridad en el modelo físico
  - Ricardo integra análisis de riesgos en el documento final

## 2. Demo de la base de datos implementada

Presentación realizada por Mihael y Miguel:

- **26 tablas creadas correctamente**
- Todas las **constraints** implementadas
- **Índices** agregados en columnas clave para optimizar consultas
- **Tablespace “BiblioGuest”** configurado y funcionando
- La base de datos muestra estructura estable y lista para pruebas

# Informe Técnico - Sistema BiblioGuest

## 1. Presentación Técnica

El sistema **BiblioGuest** fue diseñado e implementado utilizando **Oracle Database**, aprovechando su robustez, estabilidad transaccional, seguridad y capacidades avanzadas de PL/SQL.

El objetivo de este documento es describir, desde una perspectiva técnica, la arquitectura de la base de datos, su estructura interna, los modelos utilizados, la lógica de negocio implementada, y los objetos desarrollados para garantizar un funcionamiento eficiente, seguro y escalable del sistema.

Oracle Database XE 21c fue seleccionado porque:

- Es una base de datos **ACID** de grado empresarial, lo que garantiza confiabilidad en operaciones como préstamos y reservas.
- Permite el uso de **PL/SQL**, un lenguaje altamente optimizado para lógica dentro de la base.
- Incluye mecanismos nativos de control de concurrencia, integridad referencial, particionamiento y seguridad.
- Es estándar en entornos corporativos y universitarios.

## Arquitectura del sistema

BiblioGuest implementa una arquitectura de tres niveles con una clara separación entre las capas de presentación, aplicación y datos.



## 2. Objetivos Técnicos del Trabajo Final

### 2.1 Modelar correctamente los procesos clave de una biblioteca universitaria

La estructura de la base de datos debía representar con exactitud:

- Préstamos
- Reservas por horarios
- Equipos físicos (laptops, cubículos, ejemplares)
- Reglas operativas y sanciones
- Jerarquías institucionales (áreas → unidades → bibliotecas)

Esto requiere un **modelo conceptual bien definido**, con cardinalidades claras y entidades normalizadas.

## 2.2 Implementar una BD consistente, normalizada y transaccional

Se aplicaron:

- **Reglas de normalización hasta 3FN**, evitando duplicidad.
- **Constraints PK/FK/CHECK** para asegurar la integridad.
- **Transacciones controladas** mediante PL/SQL para evitar inconsistencias cuando múltiples usuarios realizan reservas simultáneas.
- **Bloqueos optimistas usando SELECT FOR UPDATE** en procedimientos críticos (evita doble reserva del mismo recurso).

## 2.3 Garantizar integridad de datos mediante reglas y PL/SQL

La BD no depende de la aplicación para validar reglas. Usamos:

- **Triggers** para mantener estados válidos (ej. evitar préstamo si el ejemplar no está “disponible”).
- **Funciones** que calculan disponibilidad o vigencia.
- **Procedimientos** que centralizan la lógica y evitar duplicación en el backend.

Esto asegura que un error o bypass en la aplicación **no puede corromper la información**.

## 2.4 Proveer una base robusta para futuras aplicaciones

El diseño modular facilita:

- Integración con aplicaciones móviles o web.
- Extendibilidad: agregar reservas de auditorios, equipos audiovisuales, etc.
- Auditorías, reportes e interoperabilidad con sistemas de la UNMSM.

# 3. Resumen de Funcionalidades, Alcances y Limitaciones

## 3.1 Funcionalidades cubiertas

- **Registro y administración de áreas, unidades académicas y bibliotecas**

Esto permite modelar la estructura institucional de la universidad.

Cada biblioteca depende de una unidad académica, y esta de un área.

- **Gestión de usuarios y grupos**

Se diferencian usuarios individuales y grupos de estudio (para cubículos).

La asociación mediante tabla puente permite flexibilidad para futuros servicios.

- **Catálogo bibliográfico completo**

El sistema guarda libros, autores, categorías, etiquetas y ejemplares.

Se permite multi-etiquetado (N-N), lo que facilita búsquedas avanzadas.

- **Gestión de laptops y cubículos**

Ambos recursos tienen

- Estados
- Bibliotecas asociadas
- Información técnica o de capacidad

- **Procesos de préstamo y reserva**

Con reglas como:

- Duración máxima
- Horarios válidos
- Disponibilidad
- Sanciones

- **Gestión de sanciones**

Automáticas y manuales, asociadas a usuarios, con fechas de inicio y fin.

### **3.2 Alcances**

- La BD está totalmente **transaccional**: si ocurre un error, nada se confirma.
- Contiene **datos de prueba** que permiten testear todos los casos.
- Usa **procedimientos, triggers y constraints** para garantizar coherencia interna.

### **3.3 Limitaciones**

- **Sin auditoría nativa o personalizada**
- **El sistema no gestiona pagos ni cobros de multas**
- **No existe gestión avanzada de inventarios (auditoría física del stock / garantías )**
- **Sin backup/restore automatizado**  
El respaldo deberá configurarse para producción mediante RMAN o EXPDP.

## **4. Procesos de Negocio**

Cada proceso está reflejado en:

- Tablas
- Lógica PL/SQL
- Reglas de negocio
- Restricciones

### **Préstamo de libro. Implica:**

- Verificar disponibilidad del ejemplar.

- Actualizar estado de “disponible” a “prestado”.
- Registrar bibliotecario y usuario (estudiante).

En Oracle, esto se asegura con:

- CHECK de estados
- SELECT FOR UPDATE en la reserva del ejemplar
- Triggers para mantener integridad

#### **Reserva de laptop. Requiere:**

- Validación de horario
- Validación de anticipación (normas de biblioteca)
- Estado del recurso
- Control de solapamientos de horarios

#### **Reserva de cubículo. Difiere de laptop porque:**

- Involucra grupos de usuarios
- Requiere verificar capacidad
- Soporta sanciones grupales indirectas (por el líder)

#### **Asignación de sanciones. Automática por:**

- No uso de reserva
- Devolución tardía

O manual mediante el bibliotecario.

#### **Gestión de ejemplares. La integridad del estado es crítica:**

- “disponible”
- “prestado”
- “deteriorado”

Un trigger evita inconsistencias.

#### **Gestión de normas**

Las reglas de días de préstamo y anticipación están parametrizadas, evitando hardcode en la aplicación.

## **5. Reglas de Negocio**

Cada regla tiene un propósito técnico para evitar errores de uso. Ejemplos:

RN: Ejemplar con estados válidos



Oracle usa un CHECK para evitar estados incorrectos introducidos por error humano o una aplicación defectuosa.

RN: Horarios válidos

Se usa REGEXP para validar formato 'HH24:MI' antes de procesarlo.

RN: Máx. 2 reservas por usuario

Esto evita una saturación injusta de recursos.

RN: Laptops/cubículos deben pertenecer a biblioteca

Impide inconsistencias al evitar recursos “huérfanos”.

RN: Disponibilidad obligatoria

Los procedimientos PL/SQL incluyen lógica de disponibilidad y bloqueos.

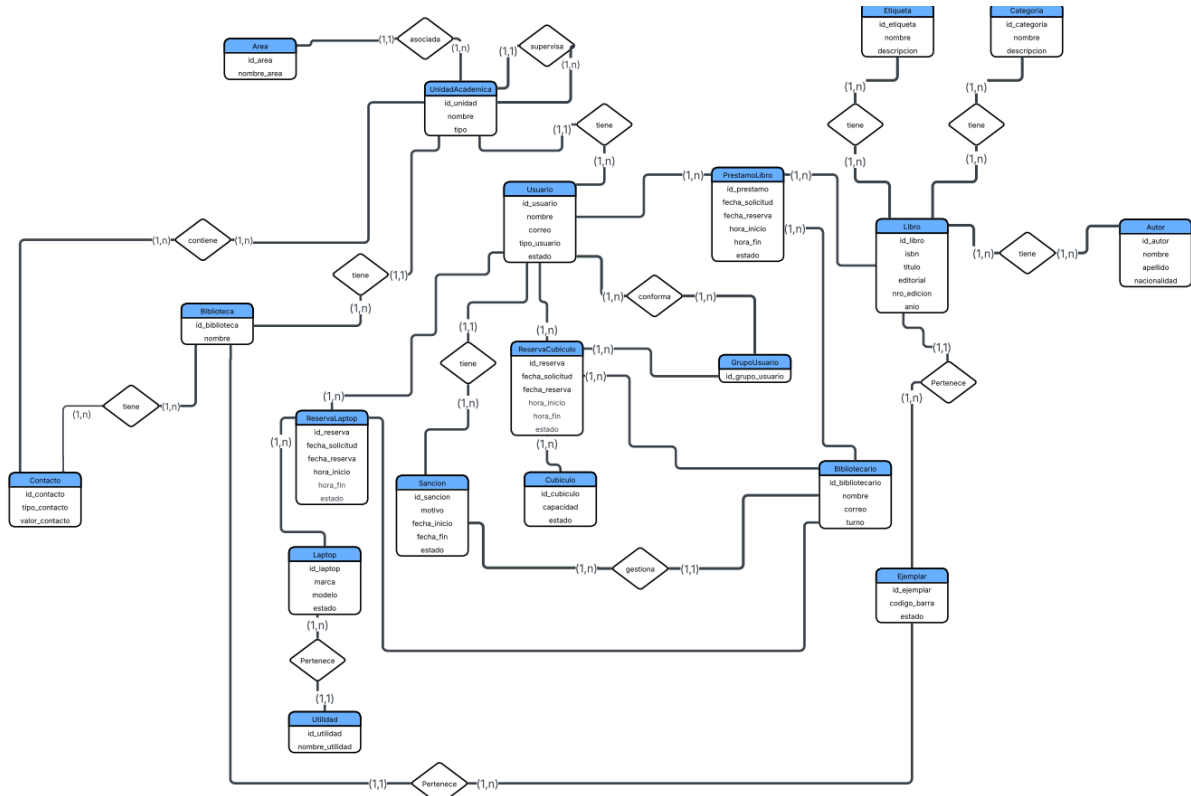
## **6. Modelo de Datos Conceptual**

Incluye aproximadamente:

- 25 entidades
- 35 relaciones
- Jerarquías: Área → Unidad → Biblioteca
- Entidades transaccionales (Prestamo, Reserva)
- Tablas puente y N-N

Uso técnico:

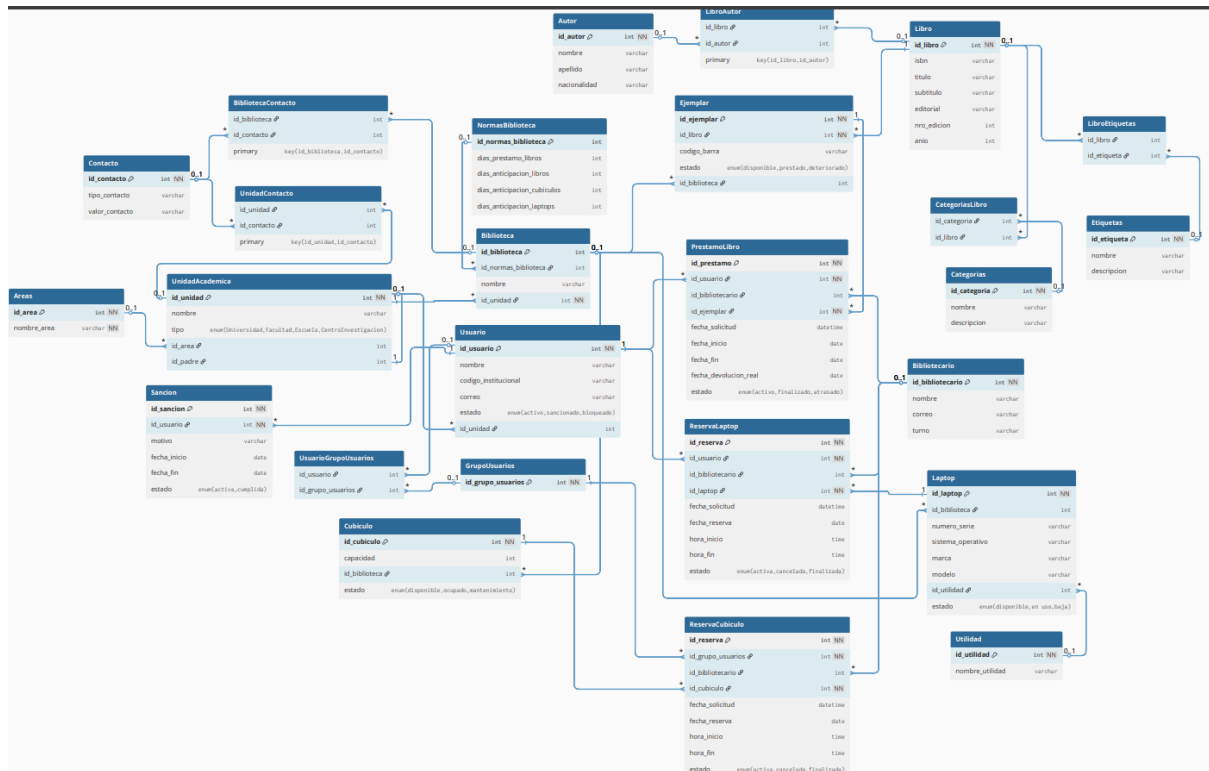
- Sirve de guía para el equipo de desarrollo
- Permite detectar redundancias antes de pasar al lógico
- Sienta una base conceptual para futuras extensiones



## 7. Modelo Lógico de Datos

Características técnicas del diseño:

- 26 tablas finales (luego de separaciones y normalización)
- Todas las claves foráneas definidas
- Uso intensivo de N-N para flexibilidad en categorización
- Tipos de datos acordes al estándar Oracle



## 8. Modelo Físico de Datos

Incluye:

- Declaración exacta de tipos Oracle (VARCHAR2, NUMBER, DATE, TIMESTAMP)
- Constraints PK, FK, UNIQUE, CHECK
- Índices estratégicos en búsquedas frecuentes

Ventajas:

- Rendimiento óptimo en consultas por libro, autor, usuario
- Integridad garantizada por el motor Oracle
- Compatibilidad con Oracle XE 21c



Validan estados, horarios, tipos.

- **Procedimientos almacenados y funciones principales**

Implementan la lógica de:

- reservar laptop
- reservar cubículo
- prestar libro
- sancionar usuario

Garantizan que la lógica no dependa del backend.

Categoría	Objetivo	Componentes clave
Funciones de utilidad	Manipulación de fecha y hora	fn_minutos,fn_build_ts
Funciones de validación	Comprobación de reglas de negocio	fn_tiene_sancion_activa,fn_reserva_solapa_laptop,fn_reserva_solapa_cubiculo
Procedimientos de transacción	Flujos de trabajo principales	pr_crear_prestamo_libro,pr_reservar_laptop,pr_reservar_cubiculo
Funciones de cálculo	Lógica financiera	fn_dias_atraso,fn_calcular_multa

- **Triggers**

Automatizan integridad como cambiar estados o registrar fechas.

Nombre del disparador	Mesa	Momento	Evento	Propósito principal
-----------------------	------	---------	--------	---------------------

trg_rl_normaliza_horas	ReservaLaptop	ANTES	INSERTAR, ACTUALIZAR	Normalizar el formato de hora a HH24:MI
trg_rl_no_solape	ReservaLaptop	ANTES	INSERTAR, ACTUALIZAR	Prevenir superposiciones de reservas
trg_rc_normaliza_horas	ReservaCubiculo	ANTES	INSERTAR, ACTUALIZAR	Normalizar el formato de hora a HH24:MI
trg_rc_no_solape	ReservaCubiculo	ANTES	INSERTAR, ACTUALIZAR	Prevenir superposiciones de reservas
trg_prestamo ajusta_estado	PrestamoLibro	ANTES	INSERTAR, ACTUALIZAR	Calcular el estado del préstamo
trg_prestamo_sync_ejemplar	PrestamoLibro	DESPUÉS	INSERTAR, ACTUALIZAR	Sincronizar el estado de la copia del libro
trg_sancion_sync_usuario	Sancion	DESPUÉS	INSERTAR, ACTUALIZAR, ELIMINAR	Sincronizar el estado del usuario

## 10. Esquema de Base de Datos Implementado (ver anexos)

### Autenticación y Autorización

- Tipo de autenticación: **token-based authentication**
- Usuarios definidos para pruebas:
  - estudiante
  - administrador
  - bibliotecario

### Roles en la Base de Datos

- Roles creados en Oracle para administración interna.

- Qué permisos tiene cada rol.
- Qué usuario del sistema los utiliza.

## Usuario principal del esquema

- Usuario del esquema: **BG\_CONNEC**
- Permisos otorgados a este usuario.

### 1. 01\_schema.sql

- **Incluye:**
  - Creación de Tablespaces físicos (BiblioGuest y BiblioGuestTemp).
  - Configuración del esquema base BG\_OWNER.
  - Definición de las tablas principales (Core) como Usuario, Libro, Bibliotecario, y sus relaciones (Foreign Keys).
- **Motivo técnico:**
  - Establece los cimientos físicos y lógicos del sistema, asegurando la integridad referencial desde el inicio.

### 2. 02\_auth\_schema.sql

- **Incluye:**
  - Modificación de tablas existentes (Usuario, Bibliotecario) para agregar la columna password\_hash.
  - Creación de la nueva tabla Administrador exclusiva para el sistema de autenticación.
  - Creación de índices específicos para búsquedas rápidas por correo durante el Login.
- **Motivo técnico:**
  - Separa la estructura de negocio de la seguridad. Prepara la base de datos para soportar autenticación robusta (JWT/Bcrypt) sin mezclar lógicas en el esquema inicial.

### 3. 03\_storeObjects.sql

- **Incluye:**
  - **Funciones:** Validaciones complejas como detección de solapes (fn\_reserva\_solapa\_laptop) y cálculo de multas.
  - **Procedimientos:** Lógica atómica de negocio (ej. pr\_crear\_prestamo\_libro, pr\_reservar\_cubiculo).
  - **Triggers:** Automatización de cambios de estado (ej. sincronizar estado de Ejemplar al prestar/devolver) y validaciones preventivas.
- **Motivo técnico:**
  - Garantiza la integridad de los datos ("Business Logic in Database"). Asegura que, aunque el Backend falle, la base de datos nunca permita un estado inválido (como una doble reserva).

### 4. 04\_security.sql

- **Incluye:**

- Creación del Rol BG\_APP\_ROLE (el "Llavero") con permisos CRUD específicos.
- Creación del usuario de conexión BG\_CONNECT (el que usa Node.js).
- Generación de **Sinónimos** para ocultar el esquema BG\_OWNER y facilitar las consultas desde el backend.
- Permisos de ejecución sobre procedimientos, funciones y vistas.
- **Motivo técnico:**
  - Implementa el principio de "Mínimo Privilegio". Protege la estructura de la base de datos evitando que la aplicación se conecte como dueño (BG\_OWNER).

#### 5. 05\_seed.sql

- **Incluye:**
  - Carga masiva de catálogos (Áreas, Autores, Libros).
  - Inserción de usuarios de prueba (Estudiantes, Bibliotecarios, Admin) con **contraseñas hashheadas reales** para pruebas de login.
  - Generación de escenarios de prueba (préstamos activos, atrasados, reservas con conflicto).
- **Motivo técnico:**
  - Provee un entorno "listo para usar" (Sandbox) que permite al equipo de desarrollo probar todas las funcionalidades (Login, Reportes, Transacciones) inmediatamente después del despliegue.

#### 6. 06\_views.sql

- **Incluye:**
  - Vistas de resumen como VW\_LIBRO\_COMPLETO (Libro + Autores + Etiquetas concatenados).
  - Vistas de detalle como VW\_PRESTAMO\_DETALLE y VW\_SANCIONES\_ACTIVAS.
- **Motivo técnico:**
  - Abstrae la complejidad de los JOINS masivos. Permite que el Backend haga un simple SELECT \* FROM Vista en lugar de escribir consultas SQL complejas de 10 líneas, mejorando el rendimiento y mantenibilidad.

#### 7. 07\_packages.sql

- **Incluye:**
  - Paquete PKG\_PRESTAMOS: Agrupa crear, cancelar, entregar y devolver libros.
  - Paquete PKG\_RESERVAS: Agrupa la gestión de Laptops y Cubículos.
- **Motivo técnico:**
  - **Organización y Encapsulamiento:** Provee una API limpia (Interfaz PL/SQL) para el Backend. En lugar de llamar procedimientos sueltos, el Backend llama a PKG\_PRESTAMOS.crear, lo que hace el código más ordenado y profesional.