



ESTUDIA EN EL  
INSTITUTO  
TECNOLÓGICO DE LAS  
AMÉRICAS (ITLA)

**1. Portada del trabajo:**

- Nombre: Sherley Nicole Peña
- Matrícula: 2023-1148
- Título del Proyecto: Sistema de Gestión de Alquiler de Locales (ITM)
- Fecha: 18/4/2025

**2. Nombre del proyecto de software:**

- Sistema de Gestión de Alquiler de Locales - ITM

**3. Tecnología para aplicar:**

- Frontend: HTML, CSS, JavaScript (jQuery)
- Backend: C# (.NET Core con Entity Framework)
- Base de Datos: SQL Server
- Plataforma de gestión: Azure DevOps

**4. Objetivo del proyecto:**

- Desarrollar una aplicación que permita registrar, visualizar y administrar el alquiler de locales por semestre en la institución ITM, facilitando el control y gestión de espacios y pagos.

**5. Alcance del proyecto:**

- El sistema cubrirá:
  - Registro de locales
  - Registro de inquilinos
  - Registro de alquileres y pagos
  - Reportes semestrales de uso y pagos
- No incluye integración con sistemas de facturación externos ni módulo de pagos electrónicos.

**6. Cronograma del proyecto:**

## **7. Definición del primer Release:**

- Registro de locales e inquilinos
- Registro básico de alquileres
- Validación de fechas y espacios

## **Requerimientos funcionales:**

- El sistema permite agregar/editar/eliminar locales
- El sistema permite registrar inquilinos
- El sistema permite asociar inquilinos con locales por semestre

## **Requerimientos no funcionales:**

- El sistema debe estar disponible al 99% en horario hábil
- Interfaz responsiva para uso en navegador moderno

## **Metodología Scrum**

### **1. Definir tareas a ejecutar:**

- Análisis de requerimientos
- Diseño de interfaz
- Desarrollo frontend
- Desarrollo backend
- Pruebas manuales y automatizadas
- Documentación y despliegue

### **2. Definir el equipo de trabajo:**

- Product Owner: Encargado de validar los requerimientos
- Scrum Master: Facilitador del proceso scrum
- Developers: Frontend y Backend
- QA Tester: Responsable de pruebas

### **3. Herramientas que usarían:**

- Azure DevOps: gestión de historias y tareas
- Visual Studio / VS Code: desarrollo
- Postman: pruebas de servicios
- Selenium: automatización de pruebas

### **4. Definir las épicas:**

- Gestión de Locales
- Gestión de Inquilinos
- Registro de Alquileres
- Reportes

### **5. Ceremonias de Scrum:**

## Historias de Usuario

ID	Historia	Criterios de Aceptación	Puntos
HU1	Como administrador, quiero registrar un local	Formulario funcional con validaciones completas	3
HU2	Como administrador, quiero ver la lista de todos los locales	Tabla ordenada con botones de acción (editar/eliminar)	2
HU3	Como administrador, quiero editar la información de un local	Edición fluida con validación de campos obligatorios	2
HU4	Como administrador, quiero registrar inquilinos	Formulario con validación de documento de identidad	3
HU5	Como administrador, quiero listar a los inquilinos	Tabla filtrable para una mejor visualización	2
HU6	Como administrador, quiero registrar un alquiler	Asociación correcta entre inquilino y local	4
HU7	Como administrador, quiero ver reportes de alquileres	Tabla agrupada por semestre y por local	3
HU8	Como administrador, quiero eliminar un local	Confirmación previa y eliminación exitosa	2
HU9	Como administrador, quiero ver los locales disponibles	Filtro por semestre que excluya los ya alquilados	3
HU10	Como administrador, quiero evitar fechas duplicadas en los alquileres	Sistema bloquea solapamientos de fechas automáticamente	4

## Plan de Pruebas

- Lista de requerimientos funcionales y no funcionales:**
  - Ver punto 7 del Release
- Criterios de aceptación y rechazo de pruebas:**
  - Aceptación: funcionalidad completada sin errores y cumple criterios de historia
  - Rechazo: errores, validaciones no aplicadas o datos incorrectos
- Herramientas de pruebas:**
  - Manual: Postman y pruebas en interfaz
  - Automatizadas: Selenium con C#
- Plantillas para casos de pruebas:**
  - Caso de prueba:
    - ID: CP01
    - Descripción: Registrar local
    - Entrada: Nombre, capacidad
    - Resultado esperado: Local registrado exitosamente
    - Resultado obtenido: \_\_\_\_\_
    - Estado: ✓/✗

5. **Equipos de pruebas y responsabilidades:**
  - QA: Pruebas manuales
  - Dev: Automatización
6. **Plan de automatización de pruebas:**
  - Casos repetitivos y críticos como registro, edición y validaciones
  - Herramienta: Selenium WebDriver con NUnit
7. **Ejecución y demostración:**
  - Capturas de pantalla y video (adjuntos por separado)

## Demostración y Entregables

1. **Video:**
  - Se adjunta video mostrando el incremento del sistema (registro de locales, inquilinos y alquileres)
2. **Links funcionales:**
  - Repositorio del código: [GitHub Link]
  - Azure DevOps con historias de usuario: [Azure Link]
  - Código de pruebas automatizadas: [GitHub/Selenium Folder Link]

## Formato del Documento

1. **Estructura del documento:**
  - Hoja de presentación
  - Índice numerado
  - Desarrollo detallado por secciones
  - Conclusión
  - Bibliografía (referencias a Microsoft Docs, Selenium, DevOps, etc.)
2. **Estilo:**
  - Tipo de letra: Calibri 11
  - Interlineado: Sencillo
  - Títulos: Calibri Light 16
  - Párrafos justificados
  - Hojas numeradas

**Entregables Finales:**

- Documento PDF
- Video de demostración
- Repositorio funcional
- Herramienta de gestión con historias activas
- Automatización funcional de pruebas