

# Proyecto #1 - LabCE

Instituto Tecnológico de Costa Rica  
Escuela Ingeniería en Computadores  
Bases de Datos (CE3101)  
I Semestre 2024  
Valor 20%



## Objetivo general

- Desarrollar una aplicación que permita manejar la descripción del caso expuesto.

## Objetivos específicos

- Aplicar los conceptos del modelo conceptual y relacional.
- Crear una Base de Datos relacional en Microsoft SQL Server para que permita el almacenamiento de los datos.
- Crear un servicio API para que centralice la funcionalidad.
- Crear una aplicación móvil utilizando SQL Lite como Base de datos empotrada.
- Crear una página Web para que exponga la funcionalidad al usuario.
- Usar herramientas como Angular, Bootstrap, HTML5, CSS, Entity Framework, y Reporting Services, Cristal Reports o similar para reportes.
- Crear un documento de instalación que permita el despliegue correcto de la aplicación en la nube.
- **Evaluar de forma objetiva, válida y precisa la solución planteada al problema complejo de ingeniería.**
- **Colaborar de forma activa en el equipo de trabajo para la realización del proyecto.**

## Descripción del problema

Gracias al aumento de la demanda de los laboratorios y activos de la empresa ha decidido crear una herramienta que permita de forma rápida y sencilla realizar la reservación de activo y laboratorios.

Se ha determinado que son requeridas las siguientes vistas:

- **Vista Administrador:** Es la vista que usaran los administradores del sistema para aspectos de configuración de los laboratorios, activos y los servicios brindados por la empresa.
- **Vista Operador:** Es la vista que usaran los operadores del sistema para realizar reservaciones, entregas y devoluciones los activos y laboratorios.
- **Vista Profesores:** Esta es la vista que permitirá a los profesores buscar activos o laboratorios y solicitar una reserva y aprobar el préstamo de activos.

# Requerimientos del Software

## Vista Administrador.

- ◆ **Log In:** Se debe contar con un 'log in' que permita autenticarse e identificar quien está ingresando (se utilizará el correo institucional como usuario y el password).
- ◆ **Gestión de laboratorios:** Permite al administrador gestionar toda la información de los laboratorios: nombre, capacidad de personas, computadores, facilidades que otorga y horario disponible y asignación de activos.
- ◆ **Gestión de Activos:** Permite al administrador gestionar toda la información de los Activos: Placa, Tipo, Marca, Fecha de Compra(Si se conoce) y si su préstamo requiere de un aprobador.
- ◆ **Gestión de Profesores:** Esta vista permitirá al administrador dar de alta, modificar o eliminar a los profesores. El Administrador debe registrar número de cédula, nombre y apellidos, edad, fecha de nacimiento, correo electrónico. Una vez registrado el sistema enviará un correo al profesor con su password (su almacenamiento será encriptado utilizando MD5).
- ◆ **Aprobación de operadores:** En esta opción el administrador podrá visualizar todas las solicitudes de registro de operadores realizadas y podrá aceptar o rechazar la solicitud. En caso de aceptarla al operador le llegara un correo electrónico indicando que ya puede utilizar el sistema.
- ◆ **Restablecimiento de password:** El sistema permitirá al administrador generar un pwd nuevo a un usuario en específico (utilizando un correo electrónico). El sistema genera aleatoriamente un pwd el cual es enviado al correo indicado y lo almacenará encriptado utilizando MD5.
- ◆ **Reportes:**

**Consulta de horas laboradas por operador:** esta opción permite al administrador visualizar por cada uno de los operadores la cantidad de horas laboradas. Debe estar ordenado alfabéticamente por Apellido1, Apellido2, Nombre.

LabCE 4/6/2024  
Escuela Ingeniería en Computadores  
Horas Laboradas por Operador

Operador	Ingreso	Salida	
<b>Perico Perico Pedro</b>			<b>5</b>
3/24/2024	8:00 AM	11:00 AM	3
3/27/2024	4:00 PM	6:00 PM	2
<b>Rivera Meneses Marco</b>			<b>10</b>
3/20/2024	7:00 AM	10:00 AM	3
3/27/2024	7:00 AM	10:00 AM	3
3/22/2024	3:00 PM	7:00 PM	4
<b>Ugalde Orozco Jaime</b>			<b>5</b>
3/20/2024	10:00 AM	12:00 PM	2
4/1/2024	7:00 PM	10:00 PM	3

## Vista Profesor.

- ◆ **Log In:** Se debe contar con un 'log in' que permita autenticarse e identificar quien está ingresando.
- ◆ **Cambio de Password:** El sistema permitirá al profesor generar un pwd nuevo. El sistema genera aleatoriamente un pwd el cual es enviado al correo indicado y lo almacenará encriptado utilizando MD5.
- ◆ **Aprobación de préstamo:** En esta opción el profesor podrá visualizar todas las solicitudes de préstamo de activos que requieran su aprobación, el sistema debe proveer la forma de aprobar o denegar el préstamo.
- ◆ **Solicitud de reservación de Laboratorio:** El sistema debe mostrar todos los laboratorios disponibles con un resumen de su capacidad, facilidades y activos.

Laboratorio	Capacidad	Computadores	Otros Activos	Facilidades	
F2-07	30	15	Proyector	Proyector hacia la pizarra y pared posterior	<a href="#">Reservar</a>
F2-08	14	14	Proyector		<a href="#">Reservar</a>
F2-09	24	12			<a href="#">Reservar</a>

El usuario selecciona un laboratorio y el sistema debe mostrar la disponibilidad de ese laboratorio para la semana actual y tres semanas hacia adelante, semana a semana.

Hoy	<	>	April 14 - 20, 2024				Laboratorio F2-02	
	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	
	14	15	16	17	18	19	20	
8 AM						+1		
9 AM				Examen Bases	Paradigmas	Examen Paradigmas		
10 AM		Introduccion	Bases De Datos		Bases De Datos		Palmda1 Julian 202218147	
11 AM								
12 PM								
1 PM		Paradigmas		Introduccion				
2 PM								
3 PM								
4 PM								
5 PM								

El profesor deberá seleccionar la fecha, hora y cantidad de horas a reservar, el sistema bloquea el espacio solicitado y envía un correo al profesor confirmando la reserva.

## Vista Operador.

- ◆ **Registro:** Esta vista permitirá a los operadores darse de alta. El operador debe registrar su número de cédula, carnet, nombre y apellidos, edad, fecha de nacimiento, correo electrónico y password (debe almacenarse encriptado usando MD5).
- ◆ **Log In:** Se debe contar con un 'log in' que permita autenticarse e identificar quien está ingresando.
  - En el caso de los operadores el inicio de un turno (login) debe quedar registrado la hora de inicio. A la hora de realizar un logout debe quedar registrada su hora de salida también.
- ◆ **Solicitud de reservación de Laboratorio por parte de un estudiante:** El sistema debe mostrar todos los laboratorios disponibles con un resumen de su capacidad, facilidades y activos.

El usuario selecciona un laboratorio y el sistema debe mostrar la disponibilidad de ese laboratorio para la semana actual y tres semanas hacia adelante, semana a semana.

El Operador deberá seleccionar la fecha, hora y cantidad de horas a reservar por el estudiante del cual se almacena Nombre, Apellidos, correo y carnet. El sistema bloquea el espacio solicitado y envía un correo al estudiante confirmando la reserva.

- ◆ **Solicitud de préstamo de activo a profesor:** El sistema debe mostrar todos los activos disponibles en ese momento.

El usuario selecciona el activo a prestar.

El sistema solicita al profesor su contraseña, la valida y en caso de ser correcta registra el préstamo.

Del préstamo se guarda el solicitante (nombre, apellidos y correo), fecha y hora de solicitud.

- ◆ **Solicitud de préstamo de activo a estudiante:** El sistema debe mostrar todos los activos disponibles en ese momento.

El usuario selecciona el activo a prestar.

En caso de ser un activo que requiere la aprobación de un profesor este registrara el préstamo, pero quedara pendiente de la aprobación del profesor.

En caso de que el profesor ya haya aprobado el préstamo. El sistema solicita al operador su contraseña; la valida y en caso de ser correcta registra el préstamo.

Del préstamo se guarda el solicitante (nombre, apellidos y correo), fecha y hora de solicitud.

- ◆ **Registro de devolución de activo:** El sistema debe mostrar todos los activos prestados en ese momento.

El usuario selecciona el activo a devolver.

El sistema solicita al operador su contraseña (con la cual certifica que el activo devuelto se encuentra en óptimas condiciones y que no sufrió ningún daño durante el préstamo y que esta completo); la valida y en caso de ser correcta registra la devolución. **Registro de Averías.** En el caso de que a la hora de que se realice una devolución el activo haya sufrido una avería esta se registra. La avería se debe asociar a un préstamo y además un detalle de la avería.

- ◆ **Reportes:**

- **Consulta de horas laboradas:** En este caso el sistema deberá mostrar el detalle del día, hora de ingreso, hora de salida y horas laboradas por el operador.

Operador	Ingreso	Salida	
Rivera Meneses Marco			10
3/20/2024	7:00 AM	10:00 AM	3
3/27/2024	7:00 AM	10:00 AM	3
3/22/2024	3:00 PM	7:00 PM	4

Impreso por marivera@estTec@ac.cr

4/6/2024 a las 1:34pm

### Aplicación móvil.

La aplicación móvil debe poder utilizarse sin conexión, por tanto, se debe manejar un base de datos local en SQL Lite, que se sincronizará con la base de datos principal. Para el proceso de sincronización es indispensable conexión con la base de datos principal. Toda la demás funcionalidad se puede realizar sin conexión y en cuanto se reestablece la conexión se sincronizan los datos.

La funcionalidad que debe desarrollarse en la App móvil es la misma expuesta en la vista profesor.

***El término gestionar corresponde a las opciones de insertar, editar, eliminar y consultar.***

## Requerimientos no funcionales del sistema

- El Sistema debe ser una aplicación web (utilizando Angular, Bootstrap, HTML5, CSS) y debe ser desplegada localmente en el **IIS**.
- La App Móvil debe ser desarrollada utilizando SQL Lite como base de datos.
- La Base de Datos debe estar en SQL Server.
- **No se** permite el uso de Procedimientos Almacenados, Vistas o Triggers. Los scripts de Base de Datos deben implementarse en la capa de datos (una librería en proyecto de C#).
- La capa de servicios (API/WebServices/Microservicios) debe estar desarrollada en C# y debe ser desplegada localmente en el **IIS**.
- El equipo de trabajo debe seleccionar a uno de sus miembros como único punto de contacto. Todas las comunicaciones y solicitudes deben ser a través de dicho punto de contacto.

## Entregables

- Manual de Usuario.
- Documento de evidencia de la solución elaborada y planificación del trabajo.
- Se deberá documentar el código fuente.
- Evidencia de uso de un manejador de código (se recomienda Github).
- Documento de instalación.
- Plan de trabajo.
  - ◆ Metas del proyecto: Describir los entregables asociados al proyecto.
  - ◆ Roles: Describir los roles y responsabilidades utilizados en el desarrollo del proyecto.
  - ◆ Reglas: Describir las reglas principales que han sido definidas para el trabajo en equipo.
  - ◆ Cronograma: Plan de proyecto con las actividades planeadas, responsables de cada actividad y fechas de entregas estimadas.
- Script de Base de Datos.
- Script de población de Base de Datos.
- Aplicación WEB.
- Web APIs.
- Minutas de las sesiones de trabajo: De manera que se evidencie la participación activa en el equipo de trabajo de acuerdo con el rol asignado y el seguimiento al plan de trabajo.

*NOTA: Cada documento solicitado debe tener la estructura de un documento técnico (portada, índice de contenidos, introducción, entre otros).*

## Documentación evidencia de solución elaborada y planificación del trabajo

- Se deberá entregar un documento que contenga:
  - ◆ Modelo conceptual utilizando la notación de Chen.
  - ◆ Modelo relacional.
  - ◆ Descripción de las estructuras de datos desarrolladas (Tablas).
  - ◆ **Descripción detallada de la arquitectura desarrollada.**
  - ◆ Problemas conocidos: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.
  - ◆ Problemas encontrados: descripción detallada, intentos de solución sin éxito, soluciones encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.
- Conclusiones del proyecto.
- Recomendaciones del proyecto.
- Bibliografía consultada en todo el proyecto.

## Aspectos operativos y evaluación:

1. **Fecha de entrega:** Se establece el siguiente plan de entregas parciales:
  - a. Plan de Trabajo: 12/Abril/2024
  - b. Resumen Ejecutivo Avance 1: 19/Abril/2024
  - c. Resumen Ejecutivo Avance 2: 26/Abril/2024
  - d. Funcionalidad completa: 4/Mayo/2024
2. El proyecto tiene un valor de **20%** de la nota del curso.
3. El trabajo es **en grupos de 5 personas**.
4. La implementación tendrá un valor de un 70% de la nota final, debe estar funcional. La defensa vale un 10% y la documentación externa un 20%.
5. Cumplir con los requerimientos especificados en la documentación no significa que se tienen todos los puntos, se evaluará que la documentación sea coherente, acorde al tamaño del proyecto y el trabajo realizado, no escatimen en documentación.
6. Cada grupo recibirá una nota en cada uno de los siguientes apartados Código, Defensa y Documentación.
7. El profesor no sólo evaluará la funcionalidad del proyecto, esto quiere decir que aunque el proyecto este 100% funcional esto no implica una nota de un 100, ya que se evaluarán aspectos de calidad de código, aplicación del paradigma **OOP**, uso de herramientas solicitadas, calidad de documentación interna y externa, trabajo en equipo.
8. No se revisarán funcionalidades parciales, ni funcionalidades no integradas.
9. Es responsabilidad de cada miembro del grupo conocer su código, el profesor puede preguntar a cualquier miembro del grupo que le explique alguna funcionalidad/porción de código.
10. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
11. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones:
  - a. Si no se entrega documentación, automáticamente se obtiene una nota de cero en el proyecto.

- b. Si no se utiliza un manejador de código se obtiene una nota de cero en el proyecto.
  - c. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de cero en el proyecto.
  - d. Si el código no compila se obtendrá una nota de cero en el proyecto, por lo cual se recomienda realizar la defensa con un código funcional.
  - e. Si el grupo no cuenta con los equipos necesarios para realizar la revisión y no avisó al profesor de esta situación obtendrá una nota de cero en el proyecto.
  - f. El código debe ser desarrollado en C#, en caso contrario se obtendrá una nota de cero en el proyecto.
  - g. Si el grupo no se presenta a la revisión/defensa se obtiene nota de cero en el proyecto. Si un estudiante no se presenta a la defensa y no cuenta con una justificación válida se le asignará al estudiante una nota de cero en el proyecto.
12. Cada grupo tendrá como máximo 50 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa y el profesor dispondrá de 10 minutos para dar la retroalimentación y la nota del proyecto.
13. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
14. Cada grupo es responsable de llevar los equipos requeridos para la revisión.
15. Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.
16. Las revisiones se realizan con los estudiantes matriculados en el curso, cualquier persona fuera de estos y los mencionados en el punto 15, no pueden participar en la revisión.



## Referencias

**AngularJS** (2018-10-04). Recuperado de: <https://angularjs.io>

**Bootstrap Themes & Templates** (2018-10-04). Recuperado de: <https://wrapbootstrap.com/>

**How to Write Doc Comments for the Javadoc Tool.** (2018-10-04). Recuperado de: <http://www.oracle.com/technetwork/articles/java/index-137868.html>

**C# Coding Conventions (C# Programming Guide).** (2018-10-04). Recuperado de: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>