**Universidad Central**

**“Marta Abreu” de Las Villas**

**Facultad Matemática, Física y Computación**

**Ingeniería Informática**



**Trabajo Extraclase Integrador de**

**Modelado y Diseño de Interfaces**

**Diseño y Programación Orientado a Objetos.**

2025

Autores:

Melissa González Oves

Chavelys Fernández Quintero

Zailenys García Rodríguez

****

Laboratorio de Computación

# RESUMEN

Hemos desarrollado un sistema de gestión para laboratorios de computación en una facultad, implementado en Java con un enfoque en programación orientada a objetos. El sistema permite administrar locales (laboratorios docentes, de proyectos y colectivos de investigación), computadoras, personas (estudiantes, profesores) y bitácoras para registrar el uso de los recursos. La arquitectura del sistema se basa en una jerarquía de clases, con clases abstractas como Local y Persona que son extendidas por clases más específicas. Esto facilita la reutilización de código.

Implementamos excepciones personalizadas para manejar errores específicos del dominio, como intentar registrar una computadora duplicada o acceder a una que no existe. La clase Bitácora permite registrar y analizar el uso de las computadoras y locales, con métodos para calcular tiempos de uso, porcentajes de aprovechamiento y generar informes. La clase Facultad actúa como punto central para gestionar todos los locales y personas, ofreciendo funcionalidades como identificar el local con mejor o peor aprovechamiento o buscar información detallada de una persona.

Índice

[RESUMEN 2](#_Toc201086084)

[INTRODUCCIÓN 4](#_Toc201086085)

[DESARROLLO 5](#_Toc201086086)

[Diagrama de caso de uso del sistema 5](#_Toc201086087)

[Descripciones textuales de cada caso de uso 6](#_Toc201086088)

[Diagrama conceptual 6](#_Toc201086089)

[Diagrama de secuencia 7](#_Toc201086090)

[Diagrama de secuencia del caso de uso **Gestionar local** 7](#_Toc201086091)

[Diagrama de secuencia del caso de uso **Gestionar PC** 8](#_Toc201086092)

[Diagrama de secuencia del caso de uso Conocer porcentaje de aprovechamiento del local 8](#_Toc201086093)

[Diagramas de clases 8](#_Toc201086094)

[Diagrama de clases del caso de uso Gestionar Local y Calcular porcentaje de aprovechamiento 9](#_Toc201086095)

[Diagrama de clases del caso de uso **Gestionar PC** 9](#_Toc201086096)

[Diagrama de clases del sistema general 10](#_Toc201086097)

[Diagramas de colaboración 10](#_Toc201086098)

[Diagrama de colaboración del caso de uso **Gestionar Local** 10](#_Toc201086099)

[Diagrama de colaboración del caso de uso **Gestionar PC** 11](#_Toc201086100)

[Diagrama de colaboración del caso de uso Calcular peor porcentaje de aprovechamiento 11](#_Toc201086101)

[Diagrama de colaboración del caso de uso Calcular mejor porcentaje de aprovechamiento 11](#_Toc201086102)

[Patrones de diseño GRASP aplicados 11](#_Toc201086103)

[Principios y reglas del diseño de interfaz de usuario 13](#_Toc201086104)

[Principios que guían el diseño de UI 13](#_Toc201086105)

[Reglas que se aplican al diseñar interfaces 13](#_Toc201086106)

[Implementación de los casos de uso 13](#_Toc201086107)

[Implementación del caso de uso **Gestionar Local** 14](#_Toc201086108)

[Implementación del caso de uso **Gestionar PC** 15](#_Toc201086109)

[Implementación del caso de uso Calcular porcentaje de aprovechamiento 16](#_Toc201086110)

[Breve explicación 16](#_Toc201086111)

[CONCLUSIONES 17](#_Toc201086112)

[RECOMENDACIONES 18](#_Toc201086113)

[BIBLIOGRAFÍA 19](#_Toc201086114)

# INTRODUCCIÓN

El problema consiste en la necesidad de gestionar el uso de las computadoras dentro de los diferentes locales de una facultad, tanto para actividades docentes como investigativas o laborales. Se requiere desarrollar un sistema con interfaz visual que permita controlar el tiempo de uso de las computadoras por parte de estudiantes y profesores, asegurando un registro detallado de la utilización de los equipos según el tipo de local y el personal autorizado para utilizarlos. Además, el sistema debe permitir monitorear el estado de cada computadora, generar reportes sobre fallas, calcular el porcentaje de aprovechamiento de los equipos y locales, así como visualizar bitácoras de uso por hora. Para ello, se deberá diseñar un diagrama de casos de uso, un modelo de clases, implementar el programa con elementos visuales y gestionar la persistencia de la información mediante archivos.

El objetivo principal de este trabajo es desarrollar un sistema que optimice el uso de las computadoras en los locales de una facultad, asegurando un control eficiente del tiempo de utilización por parte de estudiantes y profesores. Se busca gestionar adecuadamente el estado de las computadoras, identificando cuáles están ocupadas, libres o rotas, y proporcionando reportes sobre las fallas detectadas para facilitar su mantenimiento. Además, el sistema permitirá monitorear el rendimiento de los locales y equipos, calculando su porcentaje de aprovechamiento para mejorar la eficiencia en su uso. También facilitará la búsqueda de información sobre los usuarios que han empleado las computadoras y el tiempo de trabajo registrado, identificando a los más frecuentes. Otro aspecto clave es la generación de bitácoras detalladas por hora para visualizar el historial de cada equipo y local, garantizando un registro preciso. Para asegurar la disponibilidad de los datos a lo largo del tiempo, se implementará una funcionalidad de persistencia mediante archivos, de modo que la información relevante pueda ser consultada en el futuro. Finalmente, el sistema contará con una interfaz visual intuitiva que permita ingresar, procesar y visualizar la información de manera efectiva y accesible, proporcionando una herramienta confiable para mejorar la gestión de los recursos tecnológicos en la facultad.

# DESARROLLO

El problema consiste en gestionar el uso de las computadoras en los distintos locales de una facultad, asegurando el registro preciso del tiempo de utilización por estudiantes y profesores. Estos locales pueden ser de colectivos de investigación, laboratorios docentes o de proyectos, y departamentos donde trabajan los profesores del colectivo. Cada local tiene características particulares en cuanto a los usuarios que pueden acceder a sus computadoras, lo que requiere un sistema que controle la asignación de equipos y el acceso según el tipo de usuario.

El sistema debe permitir el monitoreo del estado de las computadoras, clasificándolas como ocupadas, libres o rotas, e incluir una funcionalidad para registrar el motivo de cada avería y generar reportes para el soporte técnico. Asimismo, debe proporcionar estadísticas sobre el uso de los equipos, como el porcentaje de aprovechamiento de cada computadora y local, la cantidad de dispositivos en cada estado, la identificación de los locales con mejor o peor rendimiento, y el historial de uso de cada persona con detalles sobre el tiempo de trabajo.

Además, el sistema deberá permitir visualizar las bitácoras de cada local y computadora por hora, facilitando el análisis detallado del uso de los recursos tecnológicos. Se implementará una aplicación con interfaz visual que ofrecerá un menú de navegación intuitivo para ingresar y procesar la información, proporcionando una experiencia eficiente para los usuarios. Para garantizar la persistencia de los datos, la aplicación almacenará la información en archivos, asegurando su disponibilidad para futuras consultas.

# Diagrama de caso de uso del sistema

El ***diagrama de casos de uso*** del sistema de gestión de computadoras de la facultad, describe las interacciones principales entre el autor (usuario del sistema) y las funcionalidades clave que ofrece la aplicación.

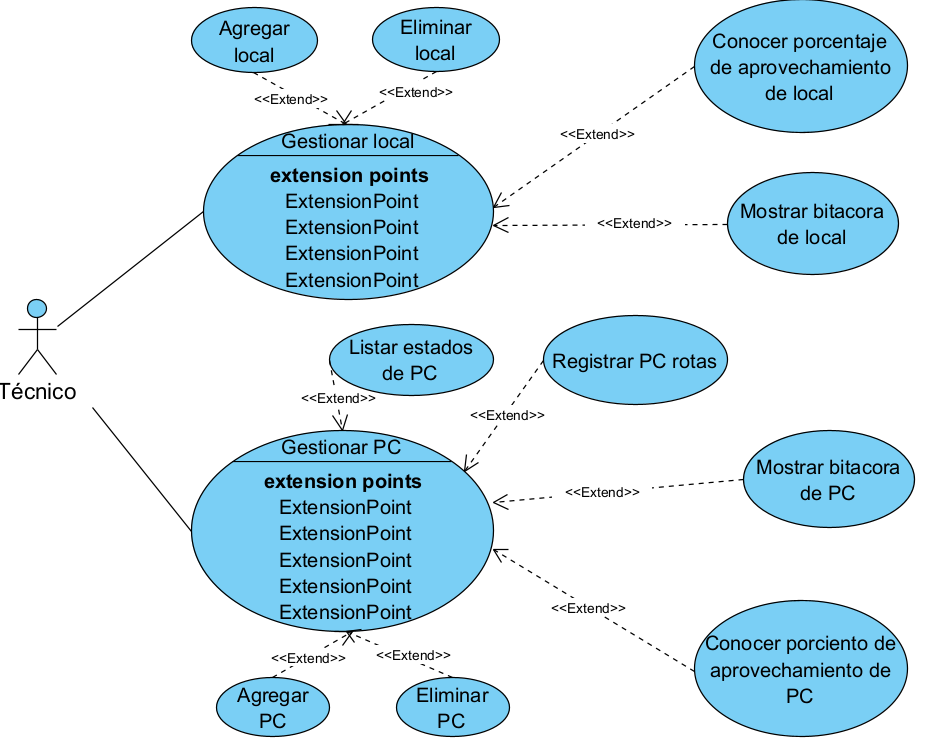


Diagrama de caso de uso del sistema

# Descripciones textuales de cada caso de uso

* Gestionar local:

Actor: Técnico.

Descripción:

1. El técnico accede al sistema y desde el menú principal, selecciona la opción ¨Locales¨.
2. El sistema muestra una tabla con todos los locales, para los cuales aparecerán botones que cumplirán las funciones de agregar, eliminar y calcular el porciento de aprovechamiento de dichos locales.
3. El técnico puede utilizar estas opciones para gestionar la información mostrada, realizar modificaciones necesarias y confirmar los cambios efectuados.
4. El sistema muestra para cada local en la tabla un botón ¨Listar estado¨, el cual es un botón que, al presionar, el sistema muestra otra tabla con el estado de cada computadora en el local seleccionado.

* Gestionar PC:

Actor: Técnico.

Descripción:

1. El técnico accede al sistema, y selecciona la opción ¨Computadoras¨ desde el menú principal.
2. El sistema muestra una tabla con todos los datos de las computadoras, para los cuales aparecerán botones que cumplirán las funciones de agregar, eliminar y calcular porciento de aprovechamiento de dichas computadoras.
3. El técnico realiza cambios y confirma.
4. El sistema muestra para cada computadora en la tabla un ¨Listar estado¨, el cual es un botón que, al presionar, se muestra otra tabla con las computadoras rotas y un ¨Reportar rotura¨, el cual es un botón que, al presionar, el sistema mostrará un formulario donde se especificará el número de la computadora y el motivo de la rotura.

* Calcular porcentaje de aprovechamiento de Local:

Actor: Técnico.

Descripción:

1. El técnico ingresa al sistema, y desde el menú principal selecciona la opción ¨Locales¨.
2. El sistema muestra una nueva ventana en la que aparecerá el botón ¨Calcular¨, con el cual, al presionar, se ingresará a otra ventana.
3. El técnico podrá elegir si calcular el mejor o peor porcentaje de cada local.
4. El técnico presionará la opción que desee y posteriormente accionará el botón ¨Calcular¨ para que el sistema realice dicha operación.

# Diagrama conceptual

El ***diagrama conceptual*** establece las entidades principales y sus relaciones, proporcionando una representación abstracta de la estructura del sistema antes de definir los detalles específicos de implementación. En este contexto, las entidades esenciales incluyen **Facultad, Local**, **Computadora**, **Profesor** y **Estudiante**.

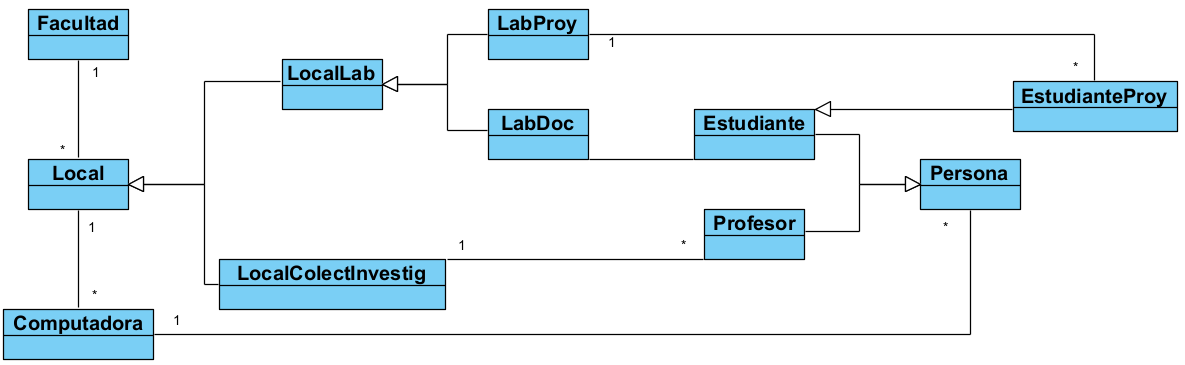


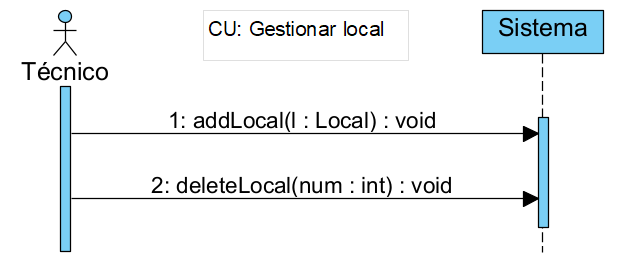
Diagrama conceptual como modelo del dominio

Este diagrama conceptual servirá como base para diseñar el diagrama de clases, permitiendo estructurar las interacciones y funcionalidades que la aplicación deberá implementar.

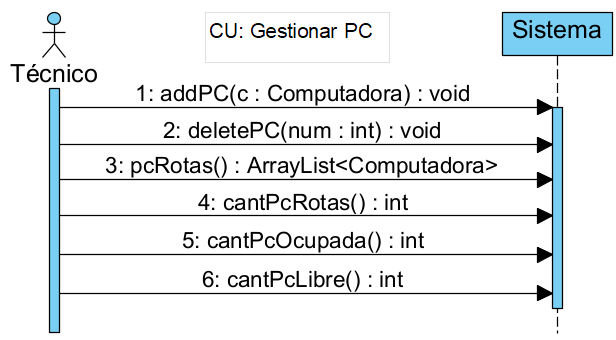
# Diagrama de secuencia

Una representación detallada de la dinámica de interacción entre los actores del sistema y sus elementos es el ***diagrama de secuencia*.** A través de estos diagramas, se describe el flujo de ejecución de los casos de uso, mostrando el orden de los mensajes intercambiados entre los objetos y cómo estos colaboran para completar una operación específica.

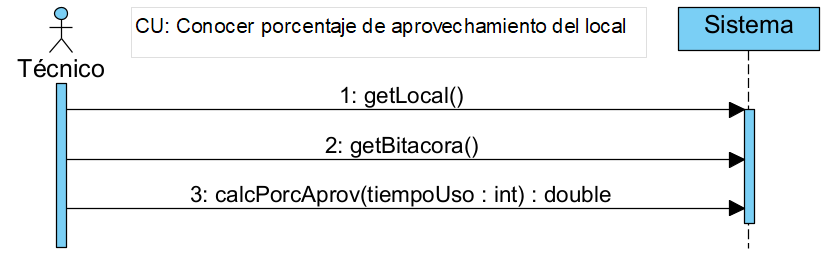
Cada caso de uso tiene un diagrama de secuencia asociado que ilustra cómo el sistema procesa las solicitudes del usuario. Estos diagramas destacan los actores involucrados, los objetos del sistema y las interacciones entre ellos, permitiendo visualizar el comportamiento interno de la aplicación. Además, ayudan a validar el diseño antes de la implementación, asegurando que la lógica de negocio se maneja correctamente.



### Diagrama de secuencia del caso de uso **Gestionar local**



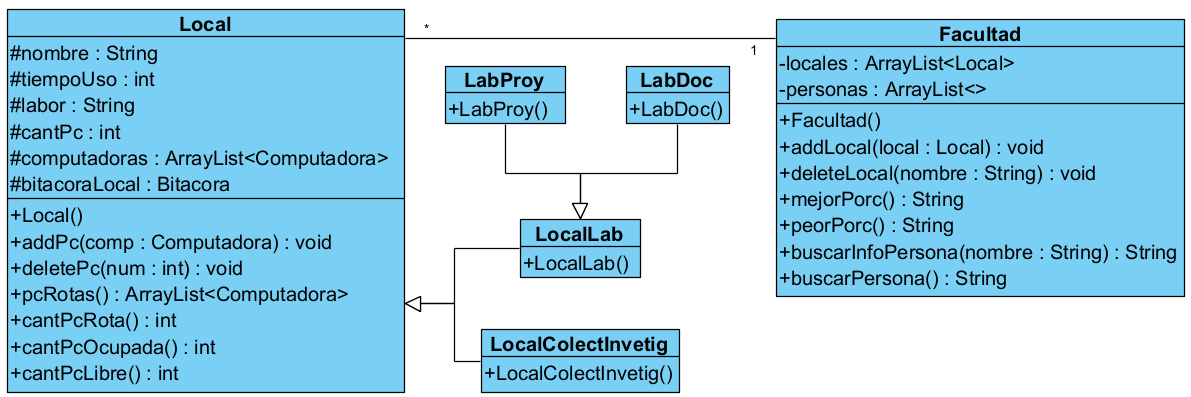
### Diagrama de secuencia del caso de uso **Gestionar PC**



### Diagrama de secuencia del caso de uso Conocer porcentaje de aprovechamiento del local

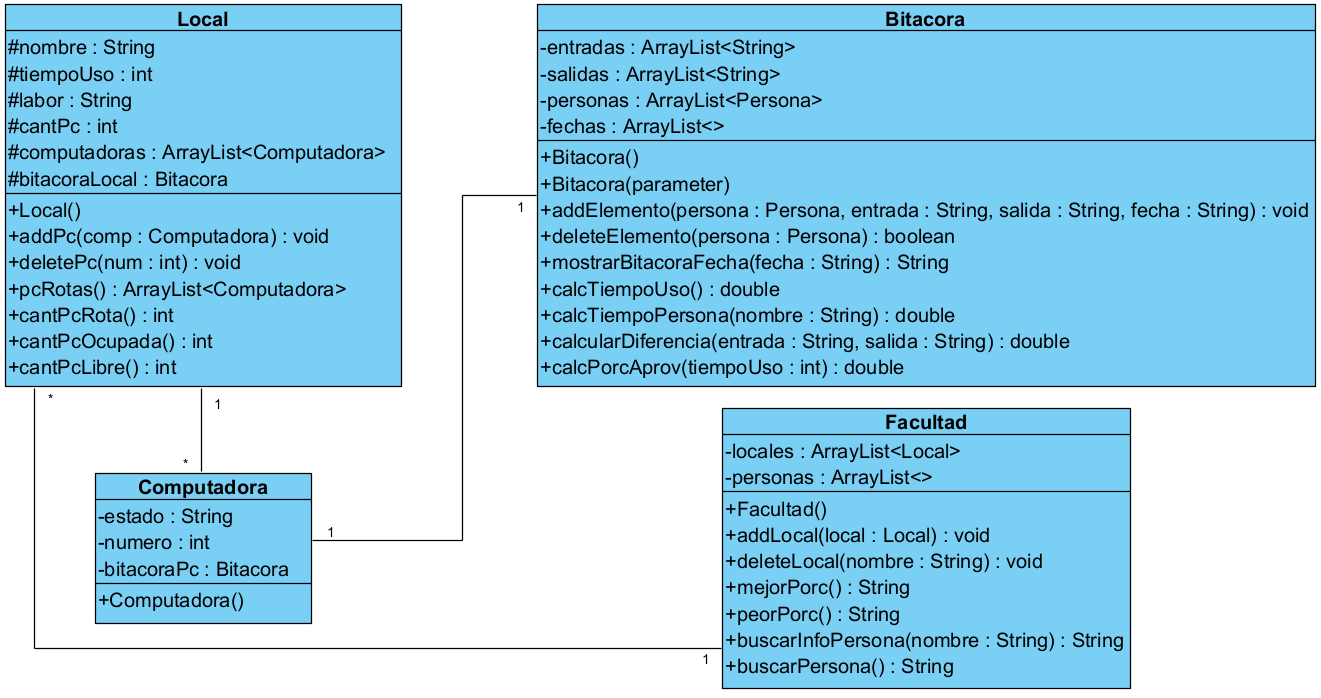
# Diagramas de clases

Para el caso de uso **Gestionar Local** y **Calcular porcentaje de aprovechamiento** las clases de manejo de datos son **Local** y **Facultad** donde esta última actúa como controlador que coordina múltiples locales y personas, siendo la principal responsable de la gestión de datos.



### Diagrama de clases del caso de uso Gestionar Local y Calcular porcentaje de aprovechamiento

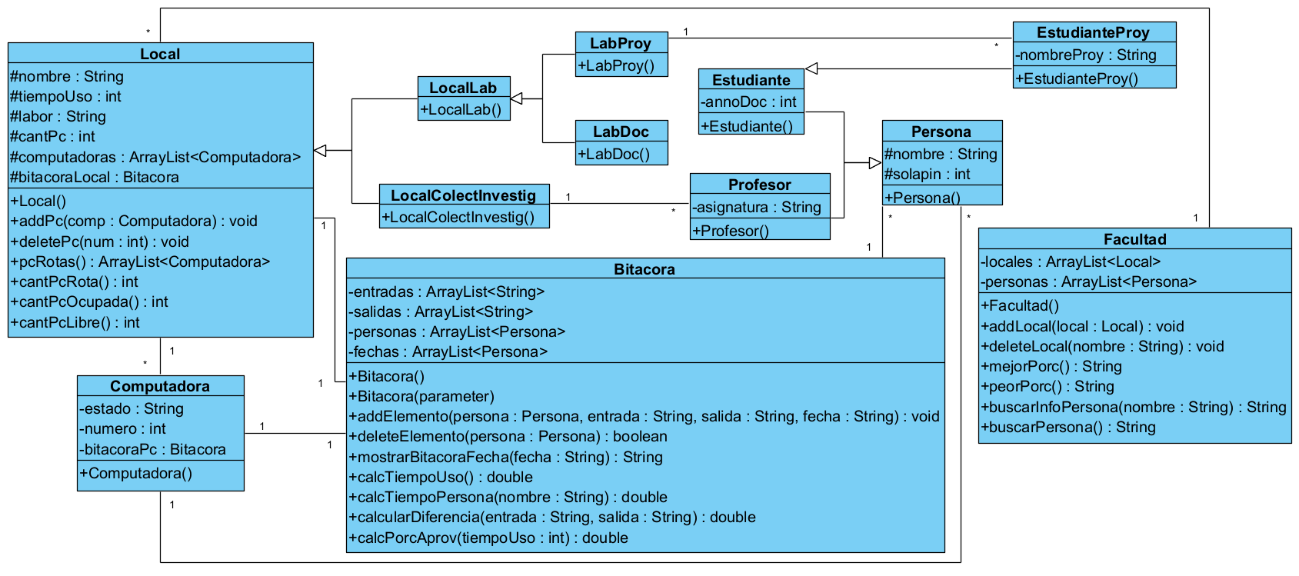
En el caso de uso **Gestionar PC** la clase **Bitacora** es la clase con la funcionalidad más robusta para manejar datos. **Facultad** actúa como el controlador que centraliza los locales.



### Diagrama de clases del caso de uso **Gestionar PC**

El ***diagrama de clases general*** del sistema de gestión de locales y computadoras está compuesto por cuatro clases principales interrelacionadas: **Local**, que gestiona espacios físicos con atributos como nombre, tiempo de uso y una lista de computadoras, junto con métodos para añadir o eliminar equipos y consultar su estado; **Computadora**, que representa cada equipo con su estado, número y una bitácora asociada; **Bitacora**, encargada de registrar actividades mediante listas de entradas, salidas, personas y fechas, con métodos para agregar registros y calcular tiempos de uso; y **Facultad**, que actúa como núcleo del sistema administrando múltiples locales y personas, con funciones para agregar o eliminar locales y generar análisis (ej. mejorPorc()).

Las relaciones clave incluyen: asociación entre **Facultad** y **Local**, y entre **Local** y **Computadora**, además de ambas con **Bitacora** para el registro de actividades. Este diseño permite una gestión modular de recursos donde **Facultad** centraliza el control, **Local** y **Computadora** manejan entidades específicas, y **Bitacora** unifica el seguimiento de eventos.

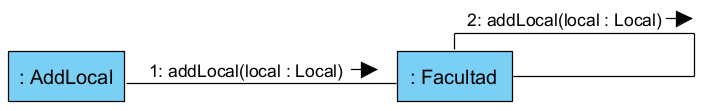


### Diagrama de clases del sistema general

# Diagramas de colaboración

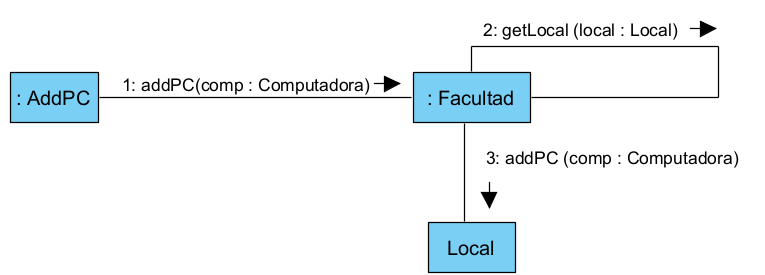
Los diagramas de colaboración son herramientas útiles para modelar interacciones entre objetos en un sistema.

El siguiente diagrama de colaboración representa la interacción entre los objetos involucrados en la operación addLocal(local: Local) dentro del sistema de gestión de locales. Este modelo resalta cómo se distribuyen los mensajes entre instancias para agregar un nuevo local a una facultad específica.



### Diagrama de colaboración del caso de uso **Gestionar Local**

A continuación, se presenta el diagrama de colaboración correspondiente al caso de uso Agregar PC en el sistema de gestión de facultades. Este diagrama muestra cómo se distribuyen los mensajes entre los objetos implicados para realizar la incorporación de una nueva computadora asociada a un local determinado.



### Diagrama de colaboración del caso de uso **Gestionar PC**

El diagrama de colaboración presentado detalla la interacción estructurada entre los objetos CalcAprovLocal y Facultad durante el proceso de cálculo de porcentajes de aprobación. Este modelo enfatiza tanto la relación estructural entre los componentes como la secuencia temporal de mensajes, representada mediante numeración explícita.

La interacción comienza cuando CalcAprovLocal envía el mensaje peorPorc()

a Facultad a través de un enlace de asociación, utilizando una llamada síncrona convencional. Una vez completada esta operación, se procede con el mensaje

mejorPorc(), manteniendo el mismo patrón de comunicación.



### Diagrama de colaboración del caso de uso Calcular peor porcentaje de aprovechamiento



### Diagrama de colaboración del caso de uso Calcular mejor porcentaje de aprovechamiento

# Patrones de diseño GRASP aplicados

El sistema implementa el patrón de diseño Experto, asignando las responsabilidades a las clases que poseen la información necesaria para ejecutar correctamente cada operación:

* La clase **Bitácora** es responsable de calcular el porcentaje de aprovechamiento, ya que contiene los registros de horarios de entrada y salida de los usuarios.
* La clase **Facultad** determina el mejor o peor porcentaje de aprovechamiento de un local, puesto que mantiene la lista completa de locales del sistema.
* La clase **Local** calcula el porcentaje de aprovechamiento de las computadoras, dado que contiene la lista de computadoras asociadas a dicho local.

De forma coherente, las operaciones de agregado y eliminación también siguen este principio:

* **Facultad** gestiona la adición y eliminación de locales, al ser quien administra su listado.
* **Local** maneja la incorporación y remoción de computadoras, como responsable de su propio conjunto de dispositivos.
* **Bitácora** se encarga de registrar y eliminar horarios, ya que almacena dicha información de forma directa.

Este enfoque asegura una correcta encapsulación de responsabilidades, permitiendo que cada clase actúe según el principio del Experto dentro del dominio que le compete.

Complementando este diseño, se aplica el patrón Creador, el cual permite distribuir de forma lógica la creación de objetos:

* La clase **Facultad** crea instancias de Local, ya que mantiene su colección.
* La clase **Local**, a su vez, es responsable de instanciar tanto su Bitácora como las Computadoras asociadas a ella.
* Finalmente, cada Computadora es capaz de generar su propia bitácora específica, garantizando independencia y trazabilidad de sus registros.

Además, el sistema incorpora el patrón Controlador, cuya función central es gestionar el flujo de información entre las clases de entidad (modelo) y las interfaces gráficas (vista). Este controlador:

* Recibe las acciones del usuario desde la interfaz.
* Se comunica con las entidades para procesar los datos y ejecutar la lógica del negocio.
* Actualiza la interfaz para reflejar fielmente el nuevo estado del sistema.

Los principios de alta cohesión y bajo acoplamiento se implementan de manera clara y funcional para lograr una estructura organizada y mantenible. La alta cohesión se evidencia en el hecho de que cada clase tiene responsabilidades bien definidas: por ejemplo, la clase Bitacora se dedica exclusivamente al manejo de registros de entradas, salidas y fechas, así como al cálculo de tiempos de uso; la clase Facultad gestiona los locales y operaciones asociadas a nivel global; mientras que la clase Local administra la infraestructura interna del local como las computadoras y su estado. Esta separación permite que cada componente evolucione sin afectar a los demás.

Por otro lado, el bajo acoplamiento se aprecia en el uso de clases que colaboran entre sí mediante métodos públicos sin depender de detalles internos. Por ejemplo, Facultad interactúa con los Local usando métodos como getNombre() o getBitacoraLocal(), sin manipular directamente sus atributos, y Bitacora se relaciona con objetos Persona de forma controlada.

# Principios y reglas del diseño de interfaz de usuario

Cuando se diseña una interfaz de usuario (UI), se aplican ciertos principios fundamentales y reglas de diseño para asegurar que la experiencia sea clara, eficiente y agradable.

### Principios que guían el diseño de UI

* **Claridad:** Cada elemento de la interfaz debe comunicar su propósito de inmediato.

*Aplicación:* Etiquetas claras en botones como “Agregar”.

* **Consistencia:** El mismo comportamiento para componentes similares en toda la app.

*Aplicación:* Todos los formularios tienen el mismo estilo de campos y botones.

* **Retroalimentación inmediata:** Mostrar al usuario que su acción tuvo efecto. *Aplicación:* Un mensaje que diga “Reporte enviado al soporte técnico”.
* **Control del usuario:** El sistema no debe sorprender al usuario.

*Aplicación:* Función “Eliminar”, mensajes de confirmación al eliminar.

### Reglas que se aplican al diseñar interfaces

* Cuanto más grande y cercano esté un botón, más fácil es hacer clic.
* Menús simples con las acciones esenciales en lugar de muchas opciones confusas.
* El usuario no debe tardar más de 3 clics en acceder a lo que busca.
* Espacios en blanco mejora la comprensión al evitar saturación visual.
* Elimina todo lo que no aporta valor directo a la tarea del usuario.

# Implementación de los casos de uso

Se muestra una parte del código correspondiente a la lógica central del caso de uso *“***Gestionar Local***”*. Este módulo se encarga de controlar las operaciones clave sobre los objetos Local, tales como la adición y eliminación de elementos de la lista de locales registrada en el sistema.

La implementación se ha diseñado con el objetivo de garantizar la integridad de los datos, evitando duplicidades y notificando al usuario en caso de intentos no válidos mediante excepciones personalizadas. Estos métodos actúan como soporte lógico para las acciones que posteriormente se desencadenan desde la interfaz de usuario, tales como los botones “Agregar” y “Eliminar” en la tabla de locales.



### Implementación del caso de uso **Gestionar Local**

A continuación, se muestra una parte del código correspondiente a la lógica central del caso de uso *“***Gestionar Computadora***”*. Este módulo implementa las operaciones fundamentales sobre los objetos Computadora, tales como la incorporación y eliminación de equipos en el sistema, el filtrado por estado y el conteo de computadoras ocupadas, libres o libres.

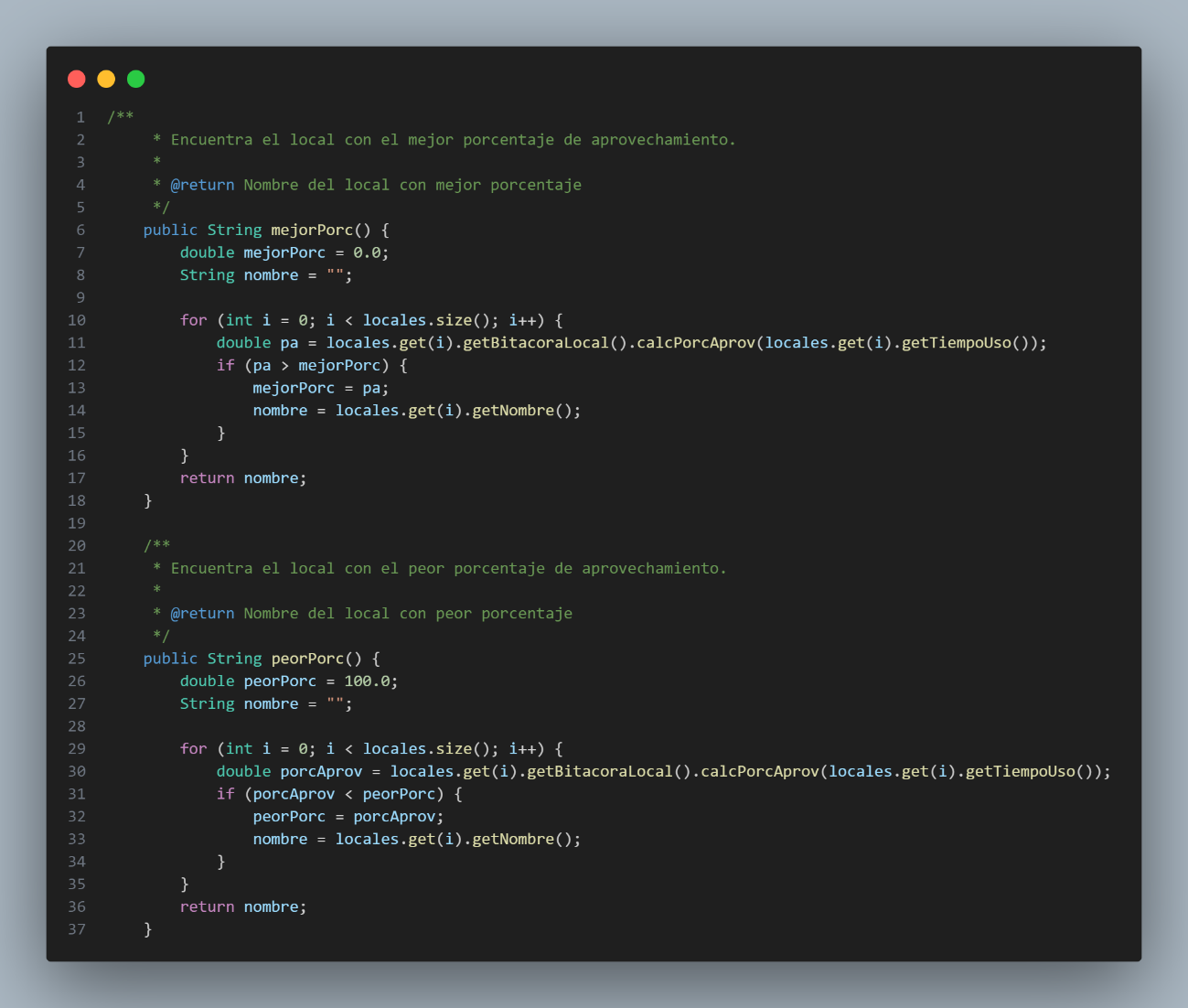
Esta implementación, al igual que en el caso de los locales, fue diseñada con el objetivo de garantizar la integridad de los datos y ofrecer un manejo estructurado de las computadoras disponibles en la facultad. Se aplican mecanismos de validación mediante excepciones personalizadas, lo que permite una retroalimentación clara ante acciones inválidas.



### Implementación del caso de uso **Gestionar PC**

Esta implementación proporciona dos funciones clave para evaluar el desempeño de locales comerciales en términos de aprovechamiento (mejorPorc(), peorPorc()).

Ambos métodos utilizan la función calcPorcAprov() de la bitácora de cada local, pasando como parámetro el tiempo disponible, para calcular el porcentaje de aprovechamiento.



### Implementación del caso de uso Calcular porcentaje de aprovechamiento

# Breve explicación

El programa es un sistema de gestión integral para laboratorios y locales de computación en entornos educativos, diseñado para registrar y analizar el uso de recursos tecnológicos. Su operación se centra en cuatro ejes principales: administración de locales (laboratorios docentes, de proyectos y colectivos de investigación), control de acceso mediante bitácoras que registran entradas/salidas de personas, análisis estadístico de tiempos de uso y generación de reportes, y persistencia de datos mediante serialización. El sistema permite calcular tiempos de utilización por persona o equipo, identificar patrones de uso, gestionar el estado de las computadoras (libres, ocupadas o dañadas), y determinar el aprovechamiento de los recursos mediante porcentajes comparativos.

# CONCLUSIONES

El desarrollo del sistema de gestión de computadoras para la facultad ha cumplido satisfactoriamente con todos los objetivos planteados, demostrando su eficacia en el control y optimización de los recursos tecnológicos. A través de una interfaz intuitiva y funcionalidades robustas, se logró implementar un mecanismo confiable para registrar el tiempo de uso por parte de estudiantes y profesores, monitorear el estado de los equipos (libres, ocupados o en mantenimiento), y generar reportes técnicos detallados. Los diagramas UML (casos de uso, clases, secuencia y colaboración) fueron fundamentales para modelar una arquitectura coherente y escalable, centrada en las entidades principales: Facultad, Local, Computadora y Bitácora.

La interfaz de usuario, diseñada bajo principios de claridad y consistencia, permite a los técnicos realizar operaciones complejas con mínimos clics, mientras que el sistema de persistencia garantiza la integridad de los datos históricos. Funcionalidades como el cálculo automático de porcentajes de aprovechamiento y las bitácoras horarias proporcionan métricas valiosas para la toma de decisiones, optimizando la distribución de recursos y reduciendo tiempos de inactividad. Además, la estructura modular del código facilita futuras ampliaciones, como la integración con sistemas institucionales o la incorporación de análisis predictivos.

# RECOMENDACIONES

**Mejoras Funcionales**

* Implementar un sistema de autenticación para garantizar acceso seguro y personalizado, integrado con los datos institucionales.
* Añadir notificaciones automáticas (correos/mensajes) para alertar al soporte técnico sobre fallas críticas en equipos.
* Incorporar visualizaciones gráficas (ej: dashboards) que muestren tendencias de uso, como horas pico o locales más demandados.
* Permitir la generación de reportes personalizados (PDF/Excel) con filtros avanzados (por fecha, local o tipo de usuario).

**Capacitación y Documentación**

* Desarrollar manuales técnicos actualizados para administradores, incluyendo diagramas de arquitectura y flujos de trabajo.
* Crear guías rápidas (con imágenes paso a paso) para usuarios finales, enfocadas en funciones clave como reportar fallas o consultar bitácoras.
* Realizar sesiones de capacitación semestrales para técnicos, cubriendo el uso avanzado del sistema y la interpretación de métricas.

# BIBLIOGRAFÍA

* *Booch, G*. La Guía del Usuario del Lenguaje Unificado de Modelado,1998.
* **Larman, C**. Aplicando UML y Patrones. Segunda edición, 2002.
* **Moreno, Rosendo**. Conferencias del curso 2017-2018. Ciencia de la Computación MFC, 2017-2018.
* **Shneiderman, B**. Las ocho reglas de oro del diseño de interfaces, mayo, 2024.