



**Benemérita Universidad Autónoma  
de Puebla**

**Facultad de Ciencias de la Computación**

**Tesis para obtener el título de:**

**Licenciada en Ciencias de la Computación**

**Administración de Bicicletas:**

**Registro de usuarios y préstamo**

**Por: María de Lourdes Montiel González**

**Director: M.C. José Alfonso Garcés Báez**

Junio 2014



## ÍNDICE

---

Agradecimiento.....	1
Resumen.....	2
Misión, Visión, Objetivos, Metas.....	3
Introducción.....	4
<b>Capítulo I. Antecedentes</b>	
1.1 El uso de la bicicleta como medio de transporte.....	6
1.2 Ventajas de la bicicleta como medio de transporte.....	6
1.3 Definiciones Técnicas.....	9
<b>Capítulo II. Descripción del problema</b>	
2.1 Propósito general.....	19
2.2 Propósitos específicos.....	19
2.3 Metodología del desarrollo de software.....	19
2.4 Metodología de desarrollo en espiral.....	20
2.5 Especificación de requerimientos.....	22
2.6 Requerimientos funcionales.....	25
2.7 Requerimientos no funcionales.....	26
<b>Capítulo III. Análisis del sistema</b>	
3.1 Diagrama contextual del módulo de registro de usuarios.....	26
3.2 Diagrama contextual del módulo de préstamo de bicicletas.....	37



3.3 Diagrama contextual del módulo de devolución .....	27
3.4 Diagrama de flujo de datos a nivel de contexto .....	28
3.5 Diagrama de casos de usos .....	30
3.6 Diagrama de clases .....	32
3.7 Escenarios .....	38
3.8 Diagrama de secuencia .....	43
<b>Capitulo IV. Diseño de la base de datos</b>	
4.1 Modelo Entidad – Relación .....	47
4.2 Diagrama Entidad – Relación .....	52
4.3 Base de Datos .....	53
<b>Capítulo V. Implementación y pruebas del sistema</b>	
5.1 Implementación .....	57
5.2 Implementación de la Interfaz .....	58
Conclusiones .....	64
Trabajo a futuro .....	65
Bibliografía .....	66
Apéndice A .....	68



## **AGRADECIMIENTOS**

---

A Dios que me ha dado fortaleza para continuar cuando a punto de caer he estado, por eso con la humildad de mi corazón le agradezco al creador de todas las cosas.

De igual forma a mis Padres que desde pequeña me enseñaron lo más importante en la vida que son los valores y que de manera incondicional siempre han estado conmigo en los buenos y malos momentos.

A mi Esposo e Hijos que han estado conmigo en este camino apoyándome y dándome ánimos a seguir adelante.

A mi familia en general, mis hermanos que han sido parte de mi vida y que me han apoyado cuando ha sido necesario.

A mis profesores de Diplomado, que con su sabiduría y experiencia supieron transmitirme los conocimientos necesarios.

A mi asesor que siempre me dio ánimos para seguir adelante, por su apoyo incondicional y la orientación que me brindo en momentos cruciales.



## RESUMEN

---

Las Aplicaciones Web, así como otras tecnologías que utilizan el Internet han tenido un crecimiento exponencial, haciendo que cada día su utilización sea ya una necesidad.

Este proyecto de tesis desarrolla una Aplicación Web, la cual permite administrar un lote de bicicletas a préstamo dentro de un campus. El presente desarrollo incluye los módulos de registro de usuarios, préstamo y devolución de bicicletas.

La Aplicación se ha probado de acuerdo a los requerimientos del administrador de bicicletas; por lo que se garantiza un buen funcionamiento del sistema. Para el desarrollo se utilizaron herramientas como WAMP SERVER y StarUML, así como Dreamweaver, que fue fundamental para un buen desarrollo.



### **Misión:**

Controlar y coordinar de manera eficiente y sistematizada un lote de bicicletas dentro de un campus que permita el fácil traslado dentro del mismo.

### **Visión:**

Tener un software portable, eficiente y eficaz, útil para la sociedad, para poder competir a nivel internacional.

### **Objetivos:**

- Tener el control de los usuarios de las bicicletas así como de los encargados de los módulos.
- Coordinar la asignación y devolución de las bicicletas.
- Fomentar la actividad física dentro del campus.

### **Metas:**

- Crear la BD de los administradores y usuarios en coordinación con la del módulo de mantenimiento.
- Diseñar el software que permita coordinar la asignación, devolución y mantenimiento de las bicicletas



## INTRODUCCIÓN

---

Los testimonios más antiguos sobre este popular vehículo (bicicleta) se remontan hasta las antiguas civilizaciones de Egipto, China e India. Pero el auténtico antecesor de la bicicleta fue el celerífero, aparato consistente en dos ruedas de madera unidas por una barra de madera de un metro de longitud, del mismo material, y que se desplazaba gracias al impulso de los pies en el suelo. El celerífero fue presentado en 1790 por el francés De Sivrac a la corte de Versalles. Casi treinta años más tarde, en 1818, el alemán Karl Von Drais unió la rueda delantera a un manillar de dirección móvil. Entre 1830 y 1840, el escocés McMillan añadió el juego de pedales y un tocoso sistema por el que se transmitía el impulso de la pedaleada a la rueda trasera. Los hermanos Michaux, en 1861, fijaron los pedales a la rueda delantera y años después incrementaron dos veces el diámetro de la misma para aumentar la velocidad de desplazamiento, había nacido el velocípedo, que fue completado por Sargent en 1865, al añadirle la cadena. En 1867 las ruedas eran metálicas; en 1875 se introdujeron las ruedas de goma maciza y en 1889 Jhon Boy Dunlop, un veterinario escocés, inventaba la cámara neumática. A partir de aquí, la estructura básica de la bicicleta estaba definida.

Siempre que se habla de la bicicleta como medio de transporte se hace énfasis en su carácter de vehículo no contaminante. Este solo hecho es tan importante, que vale por sí mismo para justificar plenamente todo el esfuerzo que se realiza para promover su uso como transporte urbano en ciudades como la nuestra. Sin embargo, además de reducir la contaminación genera muchos otros innegables beneficios como son ahorro energético, Espacio, Tiempo, Economía, Salud, Beneficios sociales etc.



Es por eso la necesidad de implementar un sistema que permita administrar un lote de bicicletas a préstamo dentro de un campus. Que incluya tres módulos, de registro de usuarios, préstamo y devolución de bicicletas.

El documento se presenta en diferentes capítulos:

El capítulo 1, hace referencia a los antecedentes, fundamentos teóricos. El siguiente capítulo muestra la descripción del problema y la especificación de requerimientos. En el capítulo 3, se hace un análisis de requerimientos del sistema y se realizan los diagramas de contexto, de uso, de clases y se describen los escenarios. El penúltimo capítulo contiene el diseño de la base de datos. En el último capítulo se desarrolla la programación, implementación y pruebas, las cuales se ilustran con la interfaz del sistema.

## 1. ANTECEDENTES

---

### 1.1 El uso de la bicicleta como medio de transporte

El uso de la bicicleta como medio de transporte en países como Holanda, Suiza, Alemania, algunas zonas de Polonia y en los países nórdicos; se ha convertido en uno de los primordiales medios de transporte. Pasando a Asia, fundamentalmente en China y La India, es la primera forma de transporte. Actualmente, en el mundo, existen cerca de 800 millones de bicicletas, siendo China el país poseedor de la mayor cantidad de estos vehículos ya que es un medio efectivo de transporte para distancias no tan extensas, es mucho más rápido que caminar, y ha traído consigo grandes beneficios a nivel de salud y de economía. [1]

En México la falta de ejercicio es la consecuencia de que ocupe el segundo lugar en la tabla de sobrepeso superado solamente por Estados Unidos y cuarto en obesidad infantil, sin embargo las mujeres mexicanas ocupan el primer lugar de obesidad en el mundo. El uso de la bicicleta es sin duda una estupenda opción como ejercicio aparte de que contribuye al ahorro, combate la contaminación y el tráfico.

Por eso es importante comenzar a formarse una cultura sobre su uso y los cambios que se verían de manera sorprendente.

### 1.2 Ventajas de la bicicleta como medio de transporte.

Algunas de las ventajas de usar la bicicleta como medio de transporte son las siguientes:

**Ecológico.** La bicicleta contribuye claramente a un medio ambiente urbano más humano, tranquilo, sano y habitable. Es silenciosa, sólo consume energía humana y no contamina. Además, el medio ambiente a nivel global



también se ve beneficiado, porque dejamos de generar una gran cantidad de gases contaminantes y contribuimos, con nuestro esfuerzo y compromiso diarios, a frenar el cambio climático.

**Eficiencia energética.** Para desplazarnos en bicicleta consumimos 50 veces menos energía que para hacerlo en coche. Por otro lado, con la energía que se emplea en fabricar un coche se pueden hacer de 70 a 100 bicicletas.

**Económico.** Al desplazarse habitualmente en bicicleta se ahorra dinero en transporte y no se gasta en gasolina, así como no se tiene que pagar por un estacionamiento.

Una bicicleta está al alcance de cualquiera. Incluso teniendo la posibilidad de alquiler o préstamo en muchas ocasiones su uso es aún más sencillo y barato. Los costes de adquisición y mantenimiento de un automóvil son 30-40 veces superiores a los de una bicicleta. La vida útil de una bicicleta (con un mínimo mantenimiento) es varias veces superior a la de un coche.

**Salud.** Moverse en bicicleta es un ejercicio de carácter aeróbico (pausado) que recomiendan médicos y cardiólogos. Si convertimos este ejercicio en un hábito nuestra salud mejorará notablemente, ya que combate el sedentarismo, aumenta el flujo sanguíneo, reduce los niveles de colesterol en la sangre, disminuye el porcentaje de grasa en el cuerpo, mejora la capacidad pulmonar Incrementa la coordinación motriz, incrementa la masa ósea: vuelve los huesos más fuertes, ayuda a mejorar el ánimo, distrae de las preocupaciones, aumenta la elasticidad, así como el rango de movimiento en las articulaciones y es un arma eficaz para combatir el sobrepeso ya que disminuye el porcentaje de grasa en el cuerpo.



La bicicleta, además, no contamina y es silenciosa, con lo cual contribuye a una mejor salud individual y colectiva al mismo tiempo. Ayuda a la salud sobre todo al sistema cardiovascular.

**Autonomía.** La bicicleta es una buena solución cuando las distancias resultan muy largas para hacerlas a pie. Es barata, accesible, fácil de manejar en todas las edades y muy recomendable para niños (mejora autoestima y autonomía) y mayores (independencia y salud).

**Seguridad.** La bicicleta en la ciudad no es un medio de transporte peligroso en sí mismo, ya que no provoca en general grandes daños y contribuye a mejorar la seguridad vial calmando el tráfico. Está demostrado que donde crece el uso de la bicicleta se reduce el número total y la gravedad de los accidentes.

**Rapidez.** En distintas comparativas entre medios de transporte realizadas en varias ciudades de nuestro país se ha comprobado que la bicicleta es, junto a la moto, la que menos tarda en desplazamientos “puerta a puerta” no superiores a 5 kilómetros, más del 50% de los desplazamientos urbanos no llegan a estos 5 km.

**Ocupación de espacio.** Cabe en la mayoría de lugares, No hay tráfico de bicicletas, en circulación como aparcadas, ocupan un espacio muy inferior al que requieren los coches. La capacidad de una vía ciclista multiplica por 10 la de una vía para automóviles. En una plaza de aparcamiento para un coche podemos estacionar entre 7 y 10 bicicletas.

**Integración social.** La bicicleta integra, porque utiliza un ritmo pausado que acerca a las personas, invita al saludo y a la conversación. El coche, al contrario, separa por su velocidad, la contaminación generada (atmosférica y acústica) y el riesgo que entraña para los viandantes y ciclistas urbanos.



**Versatilidad.** La bicicleta es fácilmente transportable (subir a un piso, caminar con ella por un parque) y permite la versatilidad, es decir, combinar varios medios de transporte tales como el autobús y el tren. Estas dos cualidades se han visto mejoradas enormemente con la aparición de las bicicletas plegables y/o fácilmente desmontables. [2]

Es por tantos beneficios la necesidad de empezar a fomentar en un Campus un proyecto tan importante como es la “ADMINISTRACIÓN DE BICICLETAS” y que posteriormente se puede implementar a otros niveles.

Se sabe de diversos intentos para sistematizar la administración de lotes de bicicletas, por ejemplo en campus como la UNAM y UDLA Puebla por mencionar algunos, sin embargo no han sido exitosos y se desconocen los antecedentes, lo cual nos obliga a recorrer nuestro propio camino.

### **1.3 Definiciones Técnicas**

Para la creación de este proyecto, se han tomado varios conceptos útiles que han brindado gran ayuda para el desarrollo de la Aplicación.

**UML.** (Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. Cubre tanto aspectos conceptuales (procesos de negocio, funciones del sistema) como cosas concretas (clases, esquemas de bases de datos, componentes reutilizables). [3]

**Diagrama.** Un diagrama o gráfico es un tipo de esquema de información que representa datos numéricos tabulados. Los diagramas siempre se utilizan generalmente para facilitar el entendimiento de largas cantidades de datos y la relación entre diferentes partes de los datos. [4]

**Flujo.** La palabra flujo se puede referir a varios conceptos de diversos campos, en el campo de la informática que es la cual nos dedicamos, tenemos:

El Flujo de trabajo es el estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas. [4]

**Dato.** El dato es una representación simbólica (numérica, alfabética, algorítmica etc.), atributo o característica de una entidad. El dato no tiene valor semántico en sí mismo, pero convenientemente procesado se puede utilizar en la realización de cálculos o toma de decisiones. En otras palabras un dato por sí mismo no constituye información, es el procesado de los datos lo que nos proporciona información como se muestra en la Fig. 1.1. [4]



*Fig. 1.1 Procesamiento de datos*

**Objeto** .Un objeto es una encapsulación genérica de datos y de los procedimientos para manipularlos. Al igual que los objetos del mundo real, los objetos de software tienen un estado y un comportamiento. El estado de los objetos se determina a partir de una o más variables y el comportamiento con la implementación de métodos. [5]

**Clase.** Una clase está formada por los métodos y las variables que definen las características comunes a todos los objetos de esa clase. [5]

**Herencia.** La herencia es un mecanismo que permite la definición de una clase a partir de la definición de otra ya existente.



La herencia permite compartir automáticamente métodos y datos entre clases, subclases y objetos. La herencia está fuertemente ligada a la reutilización del código en la OOP. Esto es, el código de cualquiera de las clases puede ser utilizado sin más que crear una clase derivada de ella, o bien una subclase.

Hay dos tipos de herencia: Herencia Simple y Herencia Múltiple. La primera indica que se pueden definir nuevas clases solamente a partir de una clase inicial mientras que la segunda indica que se pueden definir nuevas clases a partir de dos o más clases iniciales [5]

**Sistema informático.** Como todo sistema, es el conjunto de partes interrelacionadas, hardware, software y de recurso humano que permite almacenar y procesar información. El hardware incluye computadoras o cualquier tipo de dispositivo electrónico inteligente, que consisten en procesadores, memoria, sistemas de almacenamiento externo, etc. El software incluye al sistema operativo, firmware y aplicaciones, siendo especialmente importante los sistemas de gestión de bases de datos. Por último el soporte humano incluye al personal técnico que crean y mantienen el sistema (analistas, programadores, operarios, etc.) y a los usuarios que lo utilizan.

**Proyecto.** Es esencialmente un conjunto de actividades interrelacionadas, con un inicio y una finalización definida, que utiliza recursos limitados para lograr un objetivo deseado. Los dos elementos básicos que incluye esta definición son: las actividades y los recursos. Las actividades son las tareas que deben ejecutarse para llegar en conjunto a un fin preestablecido (objetivo deseado); por ejemplo: recopilar información; realizar diagnósticos; confeccionar un diseño global de un procedimiento, programar, escribir manuales de procedimiento, etc. Un aspecto fundamental en todo proyecto es el orden en el cual se realizan las actividades. Y para determinar la secuencia lógica de las actividades se debe establecer el método, el tiempo y el costo de cada operación. Los recursos son los elementos utilizados para poder realizar la ejecución de cada

una de las tareas; como por ejemplo: hardware, programas de base (sistemas operativos), programas de Aplicación, discos de almacenamiento, energía, servicios, inversiones de capital, personal, información, dinero y tiempo.

**Base de Datos.** Una base de datos (BD) se define como un “conjunto de datos relacionados entre sí”. La particularidad definitiva que convierte a un conjunto de datos en una base de datos es que la BD se controla por medio de Sistemas de Gestión de Bases de Datos (SGBDs). [6]

**Relacionados:**

- ✓ Debe existir homogeneidad en la colección de datos que conforma una BD.  
No se trata de un conjunto seleccionado de forma aleatoria.
- ✓ Los datos se recopilan y registran con una finalidad.
- ✓ Los datos deben ser relevantes con respecto a esa finalidad. [6]

**Sistema de Gestión de Bases de Datos.** Conjunto de programas de propósito general, que proporcionan funcionalidades horizontales para facilitar la gestión de la información contenida en una base de datos. Los SGBDs actúan de intermediarios entre los datos y los programas de aplicación (y sus usuarios) que los procesan y utilizan. En ocasiones, los usuarios también podrán acceder a los datos interactuando directamente con el SGBD. [6]

**Normalización.** La normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede. La normalización se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica cuando se trataban de manipular los datos.



La normalización también hace las cosas fáciles de entender. Los seres humanos tenemos la tendencia de simplificar las cosas al máximo. Otra ventaja de la normalización de base de datos es el consumo de espacio. Una base de datos normalizada ocupa menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un mucho menor uso de espacio en disco.

### **Grados de normalización.**

Existen básicamente tres niveles de normalización mayormente utilizados en las prácticas empresariales: Primera Forma Normal (1NF), Segunda Forma Normal (2NF) y Tercera Forma Normal (3NF). Cada una de estas formas tiene sus propias reglas. [7]

**Primera Forma Normal.** La Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas. Poner la base de datos en la Primera Forma Normal resuelve el problema de los encabezados de columna múltiples. [7]

**Segunda Forma Normal.** La Segunda Forma Normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la llave primaria de la tabla para identificarlos. [7]

**Tercera Forma Normal.** Una tabla está normalizada en tercera forma normal si todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas. Comentamos anteriormente que una dependencia transitiva es aquella en la cual existen columnas que no son llave que dependen de otras columnas que tampoco son llave. Cuando las tablas están en la Tercera Forma Normal se previenen errores de lógica cuando se insertan o borran registros. [7]

**PHP.** Es un lenguaje interpretado del lado del servidor que se caracteriza por su potencia, versatilidad, robustez y modularidad. Los programas escritos en PHP



son embebidos directamente en el código HTML y ejecutados por el servidor Web a través de un intérprete antes de transferir al cliente que ha solicitado un resultado en forma de código HTML puro. Al ser un lenguaje que sigue las corrientes open source, tanto el intérprete como su código fuente son totalmente accesibles de forma gratuita en la red. [8]

**Lenguaje de definición y manipulación de datos.** Un lenguaje de Manipulación de Datos (Data Manipulation Lenguaje (DML)) es un lenguaje proporcionado por el sistema de gestión de bases de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

El lenguaje de manipulación de datos más popular hoy en día es SQL, usado para recuperar y manipular datos en una base de datos relacional. Otros ejemplos de DML son los usados por bases de datos IMS/DL1, CODASYL u otras.

Se clasifican en dos grandes grupos:

Lenguajes de consulta procedimentales.

Lenguajes de consulta no procedimentales

**SQL.** Es el lenguaje estándar para trabajar con bases de datos relacionales y es soportado prácticamente por todos los productos en el mercado. Originalmente, SQL fue desarrollado en IBM Research a principios de los años setenta; fue implementado por primera vez a gran escala en un prototipo de IBM llamado System R. y posteriormente en numerosos productos comerciales de IBM y de muchos otros fabricantes. [3]

**MySQL.** Es un gestor de base de datos sencillo de usar e increíblemente rápido, también es libre y su código es abierto. Esto significa que todo el mundo puede acceder al código fuente (el código de su programación). MySQL es una herramienta completa y ha conseguido superar a una gran cantidad de bases



de datos comerciales, es un sistema de administración de bases de datos relacional, es un programa capaz de almacenar una enorme cantidad y variedad de datos, distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales o grandes empresas y organismos administrativos. Esta herramienta incluye todos los elementos de usuario, así como para administrar el sistema, proteger o hacer volcados de datos. Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados actualmente y ejecutarlo en la mayoría de los sistemas operativos. MySQL utiliza el lenguaje de consulta estructurado (SQL), este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos. [9]

**Software para diseño de páginas web** .Para hacer el diseño de una página Web, el desarrollador se puede apoyar en muchos programas existentes, ya sea para dibujos, para fotografías, para efectos especiales, en fin para diferentes utilidades que sirven para el sistema a desarrollar, entre algunos de estos programas están los de Adobe como Photoshop, Flash, Dreamweaver, Illustrator, etc.

**Dreamweaver** .Macromedia Dreamweaver 8 es un editor HTML profesional para diseñar, codificar y desarrollar sitios, páginas y aplicaciones Web. Tanto si desea controlar manualmente el código HTML como si prefiere trabajar en un entorno de edición visual, Dreamweaver le proporciona útiles herramientas que mejorarán su experiencia de creación Web. Las funciones de edición visual de Dreamweaver permiten crear páginas Web de forma rápida, sin escribir una sola línea de código. Puede ver todos los elementos o activos del sitio y arrastrarlos desde un panel fácil de usar directamente hasta un documento. Puede agilizar el flujo de trabajo de desarrollo mediante la creación y edición de imágenes en Macromedia Fireworks o en otra Aplicación de gráficos y su

posterior importación directa a Dreamweaver. Dreamweaver también contiene herramientas que facilitan la adición de activos de Flash a las páginas Web. [10]

**Estilos CSS.** Con la idea tradicional de elaboración de páginas Web, el desarrollador realiza la página Web y posteriormente añade las etiquetas para definir la apariencia de la misma. Sin embargo, al utilizar los parámetros para las etiquetas resulta tedioso modificar la apariencia de algún elemento pues es necesario actualizar todas las páginas que integran el sitio. [11]

**Flash.** Es una tecnología para crear animaciones gráficas vectoriales independientes del navegador y que necesitan poco ancho de banda para mostrarse en los sitios Web. La animación en Flash se ve exactamente igual en todos los navegadores, un navegador sólo necesitan un plug-in para mostrar animaciones en Flash.

Con Flash los usuarios pueden dibujar sus propias animaciones o importar otras imágenes vectoriales.

Flash era conocido como FutureSplash hasta 1997, cuando Macromedia Inc. compró la compañía que lo desarrolló.

**Photoshop.** Adobe Photoshop es el programa de edición de imágenes más utilizado en la industria. Es un potente programa con diversas funciones para una gran variedad de usuarios como fotógrafos, ilustradores, diseñadores gráficos, artistas plásticos, científicos y muchos otros profesionales.

Photoshop pertenece al grupo de programas de edición de imágenes, también llamados programas de mapas de bits o "Paint", que tienen un entorno basado en píxeles. Photoshop es principalmente un programa de edición de mapas de bits con ciertas funciones de texto y gráficos vectoriales. [12]

**Illustrator.** Adobe Illustrator, es el nombre o marca comercial oficial que recibe uno de los programas más populares de la casa Adobe, junto con sus programas hermanos Adobe Photoshop y Adobe Flash, y que se trata esencialmente de una Aplicación de creación y manipulación vectorial en forma

de taller de arte que trabaja sobre un tablero de dibujo, conocido como "mesa de trabajo" y está destinado a la creación artística de dibujo y pintura para Ilustración (Ilustración como rama del Arte digital aplicado a la Ilustración técnica o el diseño gráfico, entre otros). Es desarrollado y comercializado por Adobe Systems Incorporated y constituye su primer programa oficial de su tipo en ser lanzado por ésta compañía definiendo en cierta manera el lenguaje gráfico contemporáneo mediante el dibujo vectorial.

**Vías reservadas.** Son aquellas vías ciclistas que no están segregadas del tráfico, sino que ofrecen una interacción con el resto de vehículos según la lógica del tráfico, es decir, que se comporta como otro carril más de tráfico (de la misma anchura y de uso compartido varios metros antes de cada cruce) o como una calle o carretera independiente que se cruza con otra. Normalmente, debido a las características propias de la trama urbana, son de trayectoria independiente al itinerario del resto del tráfico. Por ello, las vías ciclistas reservadas se suelen encontrar en las zonas interurbanas o en grandes parques urbanos. Ejemplos de vías reservadas en zona interurbana son las vías verdes (en los que se permite, además del tránsito de peatonal, la circulación de bicicletas, pero no la de automóviles, ni en ocasiones, otros vehículos de motor) y, en las zonas urbanas, las vías ciclistas a través de grandes parques. [13]

**Vías segregadas.** Son aquellas que transcurren a lo largo de una ruta en la que también circulan vehículos de motor (carril-bici) o peatones (acera-bici) y que pretenden delimitar una porción específica del ancho de la vía para las bicicletas. [13]

**Vías integradas (calles compartidas).** Son aquellas vías en las que los usuarios de bicicleta y los de vehículos de motor circulan por el mismo espacio, de acuerdo con las normas del tráfico comunes a todos los vehículos. Las vías integradas pueden recibir un tratamiento propio (señalización específica, calmado de tráfico, etc.) para facilitar la circulación de las bicicletas. [13]

La Aplicación de vías integradas se pone en práctica cuando las calles no son suficientemente anchas para la continuidad de un carril asignado para bicicletas, los ciclistas les es permitido usar legalmente las calles, pero acorde a un plan regulado en el que ambos puedan compartir la vías. [13]

**Senderos para bicicletas.** Los senderos para bicicletas (bike trails en inglés) están completamente separados de las calles. Estos caminos que siguen sus propias rutas dentro de las ciudades, o con mayor frecuencia en las zonas rurales, se crearon a partir de rutas, carriles y caminos de arrastre de canales abandonados. La mayoría tienen fines recreativos, aunque en algunos países, como los Países Bajos y Gran Bretaña, forman parte de una red de transporte que une ciudades. La mayoría de los senderos rurales son compartidos con peatones. [13]

**Ciclovías recreativas.** Las ciclovías recreativas o «calles abiertas» (Open Streets o Ciclovía) es un movimiento internacional por la cual las calles son libres de vehículos motorizados que permiten, durante varias horas del día, principalmente los domingos y días festivos, el paseo gratuito y seguro en bicicleta, a pie o en patines, de miles de personas. [13]

**Seguridad.** Se ha hecho a menudo la distinción entre la seguridad subjetiva que las vías ciclistas proporcionan a los usuarios, y su seguridad objetiva. Mientras que las vías reservadas de diseño correcto mejoran la seguridad y el confort de los ciclistas, las vías segregadas urbanas introducen irregularidades en el flujo del tráfico que aumentan las demandas de atención y la posibilidad de errores tanto en ciclistas como en automovilistas y por tanto aumentan de hecho el riesgo y el número de accidentes de ciclistas. [13]



## **2. DESCRIPCIÓN DEL PROBLEMA**

---

Se requiere un sistema para un campus, donde se pueda administrar el uso de la bicicleta y que incluya el registro de usuarios, préstamo y devolución, así como el mantenimiento de bicicletas que incorpora al sistema Marco Antonio Espinosa García. [19]

### **2.1 Propósito General**

Diseñar y programar 3 módulos del Sistema para la Administración de Bicicletas a préstamo en un campus, mediante la metodología de espiral y desarrollo de prototipos.

### **2.2 Propósitos Específicos**

1. Diseñar la base de datos de usuarios
2. Programar el módulo de registro de usuarios
3. Programar el módulo de préstamo de bicicletas
4. Programar el módulo de devolución de bicicletas

### **2.3 Metodología de Desarrollo del Software**

Una metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software. Puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo.

Finalmente dependerá de la metodología utilizada los productos del proyecto, por esta razón es necesario, conocer a fondo cada una de ellas y poder diferenciar entre una y otra, para de este modo saber elegir la correcta en el

momento de desarrollar un nuevo software, de otra manera el producto no será el mejor e incluso puede ser inútil. [14]

Para el desarrollo del sistema “Administración de Bicicletas” Registro de Usuarios y Préstamo que se desea implementar se utilizará la metodología en espiral por las ventajas que se describen a continuación:

#### **2.4 Metodología de desarrollo en espiral**

La metodología de desarrollo en espiral es una evolución de método clásico en cascada (Waterfall, top-down) y se considera un método de desarrollo incremental donde se conjuga la naturaleza de construcción de prototipos con los aspectos controlados y sistemáticos del MODELO LINEAL y SECUENCIAL. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software que no se basa en fases claramente definidas y separadas para crear un sistema.

Este tipo de metodología equivale al de cascada, pero en él se permite el solapamiento de varias etapas con el objetivo de flexibilizar y compensar el tiempo de desarrollo total y alcanzar resultados funcionales en etapas tempranas. [15]



*Fig. 2.4 Metodología de desarrollo en espiral*

Está considerado como un método de desarrollo rápido y eficiente.

Es adecuada para proyectos en los que se tienen claros los objetivos finales pero no todos los detalles de implementación están elucidados.



La metodología de desarrollo en espiral permite construir aplicaciones de tamaño medio manteniendo los recursos constantes.

Normalmente el proyecto se divide en módulos más pequeños y a cada uno de ellos se le aplica el siguiente proceso:

**Análisis de requerimientos** -Durante esta etapa se estudia detalladamente los requerimientos que cada objetivo conlleva. Aquí establecen todos los detalles funcionales deseados.

**Diseño del sistema** - Con los datos de la etapa anterior, se diseña el sistema. Se realiza el diseño de la base de datos (en caso de ser aplicable), interface de usuario, entorno, etc.

**Etapas de construcción** - La etapa de construcción comprende básicamente la codificación y test de unidades. Esta etapa es un trabajo de programación pura.

**Test y evaluación** - En esta etapa se realiza un test del módulo completo así como su evaluación frente al estudio de requerimientos. En muchos casos en esta etapa los usuarios finales participan de manera activa aportando información decisiva para la usabilidad del sistema. Fig.2.4.

### **Ventajas**

- El modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora.
- Como el software evoluciona a medida que progresa el proceso, el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos.
- El modelo en espiral permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.



- El modelo en espiral demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto y si se aplica adecuadamente debe reducir los riesgos antes de que se conviertan en problemas.
- En la utilización de grandes sistemas ha doblado la productividad.

### **Desventajas**

- Resulta difícil convencer a grandes clientes de que el enfoque evolutivo es controlable.
- Debido a su elevada complejidad no se aconseja utilizarlo en pequeños sistemas.
- Genera mucho tiempo en el desarrollo del sistema
- Modelo costoso
- Requiere experiencia en la identificación de riesgos

### **2.5 Especificación de Requerimientos**

La Especificación de Requisitos Software (ERS) es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software. Los casos de uso también son conocidos como requisitos funcionales. Además de los casos de uso, la ERS también contiene requisitos no funcionales (o complementarios).

Los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación (Como por ejemplo restricciones en el diseño o estándares de calidad).

Está dirigida tanto al cliente como al equipo de desarrollo. El lenguaje utilizado para su redacción debe ser informal, de forma que sea fácilmente comprensible para todas las partes involucradas en el desarrollo.



Para este sistema (Administración de Bicicletas) se requiere un sistema que contenga los módulos que se describen a continuación: Modulo de registro, Modulo de préstamo, Módulo de devolución.

### **Requerimientos del módulo de registro**

El sistema contará con tres tipos de usuarios diferentes (Usuario no registrado, usuario registrado y administrador). A continuación se mencionan los requisitos para estos usuarios.

#### **Usuario no registrado:**

<b>UNR00</b>	Darse de alta en el módulo de registro para poder adquirir una bicicleta y un ID, con la siguiente documentación:
<b>UNR01</b>	Credencial oficial vigente.
<b>UNR02</b>	Matricula la cual ya fue asignada por el campus.
<b>UNR03</b>	Comprobante de domicilio actual.
<b>UNR04</b>	Teléfono de Casa.
<b>UNR05</b>	Celular.
<b>UNR06</b>	E-mail.
<b>UNR04</b>	Huella Digital.
<b>UNR05</b>	Fotografía.
<b>UNR06</b>	Firma electrónica de carta responsiva



### Usuario registrado:

<b>UR00</b>	ID la cual ya fue asignada al registrarse
<b>UR01</b>	Forma de identificarse:( Matricula, Huella digital)

### Encargado del módulo:

<b>EM00</b>	Conectarse como administrador (ID_ Encargado del módulo).
<b>EM00</b>	Elegir registro de usuario, préstamo y devolución de bicicletas.
<b>EM00</b>	Elegir actualizar datos, altas o bajas para administrar la información.
<b>EM00</b>	Introducir los datos para dar de alta un usuario.
<b>EM00</b>	Seleccionar y actualizar los datos del registro.

### Requerimientos del módulo de préstamo

<b>RMP00</b>	Este módulo contendrá un menú principal donde se pueda dar el préstamo de bicicletas
<b>RMP01</b>	Clave del usuario la cual ya debe de estar registrada en la BD.
<b>RMP02</b>	Forma de identificarse:( Matricula, Huella digital)
<b>RMP03</b>	Clave de la bicicleta.
<b>RMP 04</b>	Registrar préstamo en la Base de Datos y grabar la fecha y la hora

## Requerimientos del módulo de devolución

En este módulo se hará la devolución de bicicletas.

<b>RMD00</b>	Identificación del usuario, por medio de ID o clave de la bicicleta préstamo de bicicletas
<b>RMD01</b>	El sistema despliega automáticamente información de la bicicleta asignada.
<b>RMD02</b>	Captura Clave de la bicicleta
<b>RMD03</b>	El Sistema libera bicicleta, y abandera registro de disponible, se marca la fecha y la hora de la entrega.

## 2.6 Requerimientos funcionales

El sistema debe realizar de acuerdo a los requisitos de usuario las siguientes funciones:

<b>RF01</b>	Acceder a la BD como encargado del módulo.
<b>RF02</b>	Dar de alta a los usuarios para gestionar el registro de usuarios, el préstamo y la devolución de bicicletas.
<b>RF03</b>	Dar de baja a los usuarios que ya no cuentan con los requisitos para el préstamo de bicicletas.

## 2.7 Requerimientos no funcionales

RNF01	Contar las terminales para ingresar al sistema.
RNF02	Tener acceso a internet y trabajar en red.
RNF03	Contar con PHP y MySQL.
RNF02	El sistema tenga un desarrollo evolutivo, escalable.
RNF03	Funcionar los 365 días del año.

## 3. ANÁLISIS DEL SISTEMA

Una vez analizados los requerimientos del sistema y haber identificado las metas globales, empezamos con el análisis del sistema desarrollando los diagramas.

### 3.1 Diagrama contextual del módulo de registro de usuario

El funcionamiento del módulo de registro de usuario se describe en la Fig.3.1



Fig. 3.1 Registro de Nuevo usuario

### 3.2 Diagrama contextual del Módulo de préstamo de bicicletas.

El funcionamiento del módulo de préstamo de bicicletas se describe en la Fig.3.2.



Fig. 3.2 Modulo de préstamo

### 3.3 Diagrama contextual del Módulo de devolución.

El funcionamiento del módulo de devolución de bicicletas se describe en la Fig.3.3.

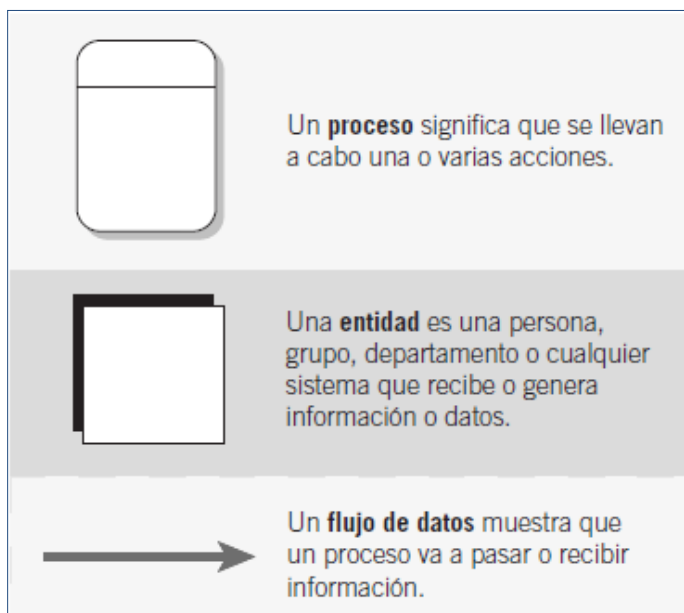


Fig. 3.3 Modulo de devolución

### 3.4 Diagrama de Flujo de Datos a Nivel de Contexto

El diagrama de flujo de datos a nivel de contexto es una manera de mostrar el alcance del sistema o lo que se va a incluir en él, es también conocido como modelo ambiental. Y se enfoca en los datos que fluyen hacia el sistema y salen de él, además del procesamiento de estos datos. Emplea sólo tres símbolos como de muestran en la Fig. 3.4 y de los cuales son:

- **Procesos** transforman los datos entrantes en información de salida y el nivel de contenido tiene sólo un proceso, que representa a todo el sistema completo.
- **Entidad** externa representa a cualquier entidad que suministra o recibe información del sistema, pero no forma parte del mismo.
- **Líneas** representan a los datos. [16]



*Fig. 3.4 Simbología del diagrama de contexto*

Para nuestro sistema se desarrolló el diagrama de contexto como se muestra en la Fig. 3.5

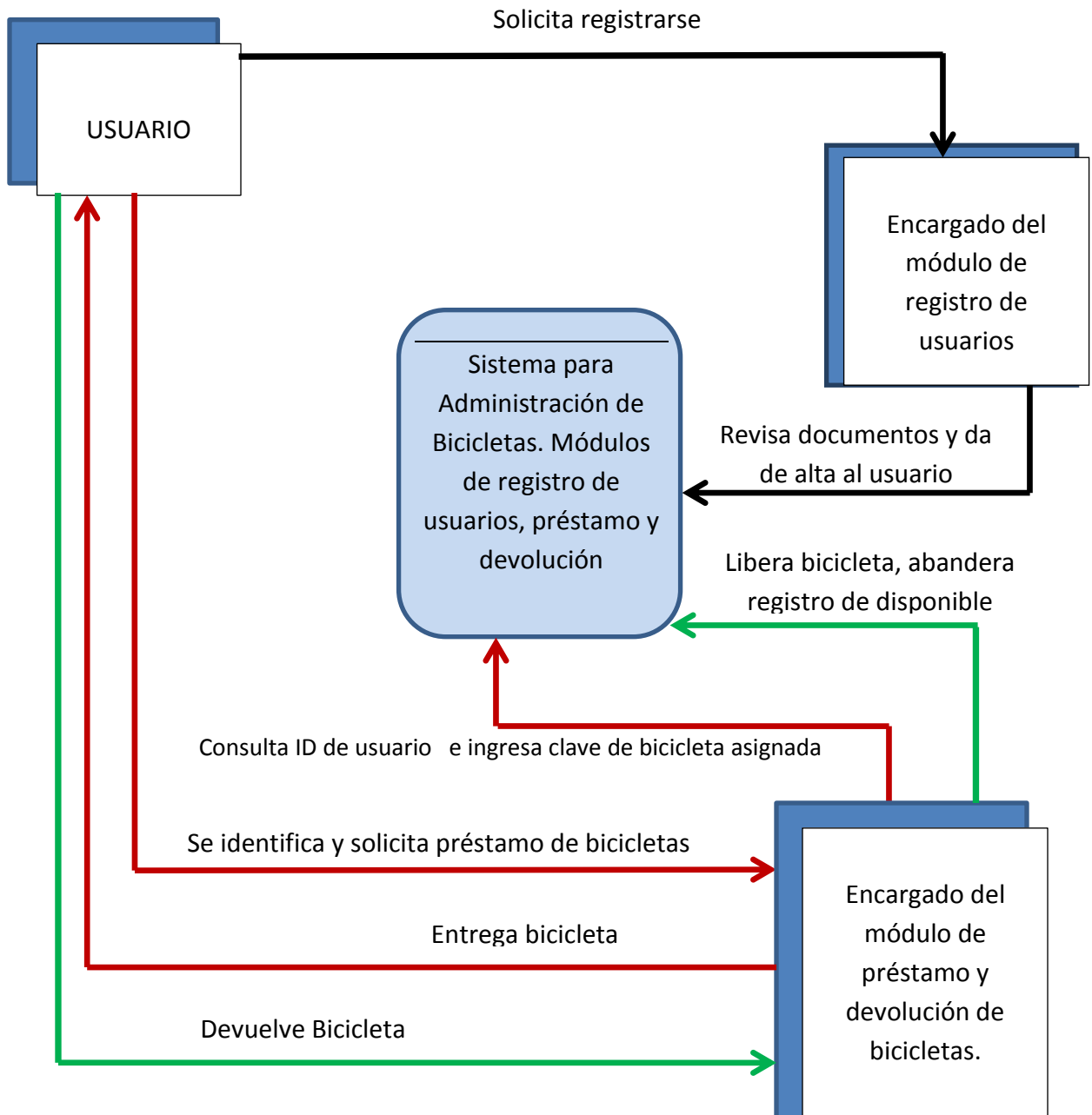


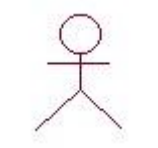


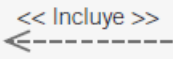
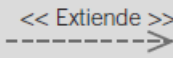

Fig. 3.5 Diagrama de contexto general

### 3.5 Diagrama de casos de usos

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos (Fig. 3.5.1)

- Actor.
- Casos de Uso.
- Relaciones de Uso, Herencia y Comunicación.

<b>Actor</b>		Un actor es una entidad externa (de fuera del sistema) que interacciona con el sistema participando (y normalmente iniciando) en un caso de uso. Los actores pueden ser gente real (por ejemplo, usuarios del sistema), otros ordenadores o eventos externos
<b>Caso de Uso</b>		Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.
<b>Relación</b>	<b>Símbolo</b>	<b>Significado</b>
<b>Comunica</b>		Para conectar un actor con un caso de uso se utiliza una línea sin puntas de flecha.
<b>Incluye</b>		Un caso de uso contiene un comportamiento común para más de un caso de uso. La flecha apunta al caso de uso común.
<b>Extiende</b>		Un caso de uso distinto maneja las excepciones del caso de uso básico. La flecha apunta del caso de uso extendido al básico.
<b>Generaliza</b>		Una "cosa" de UML es más general que otra "cosa". La flecha apunta a la "cosa" general.

*Fig. 3.5.1 Simbología del diagrama de casos de usos*

Para nuestro sistema se desarrolló el diagrama de casos de uso con la versión 10.2 de Visual Paradigm for UML Standard Edition (licencia de la BUAP). Fig. 3.5.2

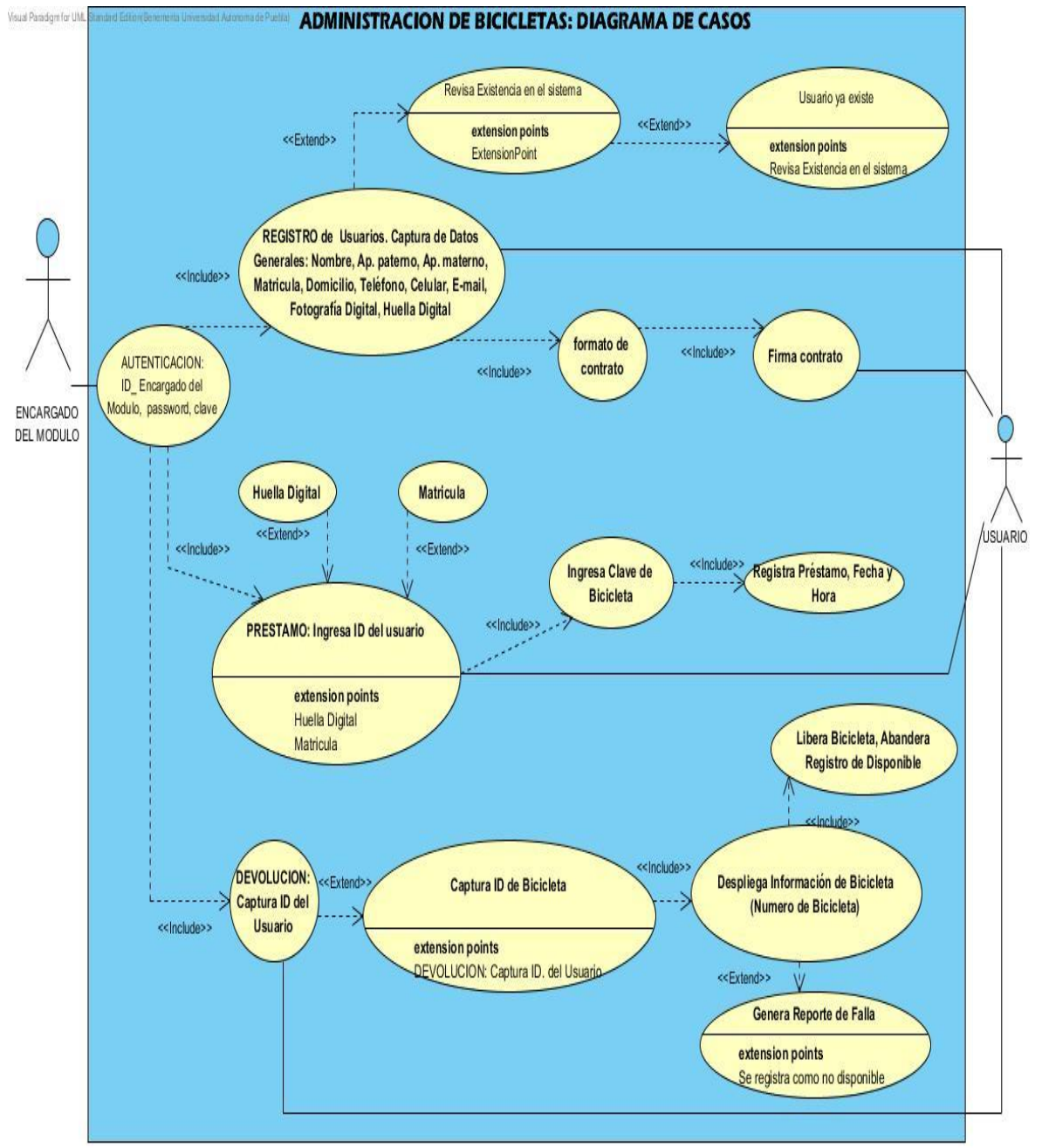


Fig. 3.5.2 Diagrama de casos de usos del Sistema Administración de Bicicletas.

### 3.6 Diagrama de clases

El diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenido.

Un diagrama de clases está compuesto por los siguientes elementos:

- **Clase:** atributos, métodos y visibilidad.
- **Relaciones:** Herencia, Asociación, Ensamblado y Uso.

#### Clase

Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, etc.). Fig. 3.6.1

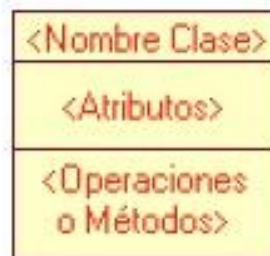


Fig. 3.6.1 Representación de una clase

En UML, una clase es representada por un rectángulo que posee tres divisiones:

En donde:

- **Superior:** Contiene el nombre de la Clase
- **Intermedio:** Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).

- **Inferior:** Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la *visibilidad*: private, protected o public).

### Atributos

Un atributo representa alguna propiedad de la clase que se encuentra en todas las instancias de la clase. Los atributos pueden representarse solo mostrando su nombre, mostrando su nombre y su tipo, e incluso su valor por defecto.

Los atributos o características de una Clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno, estos son:



**Public:** Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.



**Private:** Indica que el atributo sólo será accesible desde dentro de la clase (sólo sus métodos lo pueden acceder).



**Protected:** Indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser accedido por métodos de la clase además de las subclases que se deriven (ver herencia).

Los atributos definen la estructura de una clase y de sus correspondientes objetos. El atributo define el valor de un dato para todos los objetos pertenecientes a una clase. [17]

## Identificadores

En el momento de incluir atributos en la descripción de una clase se debe distinguir entre los atributos los cuales reflejan las características de los objetos en el mundo real, y los identificadores los cuales son utilizados exclusivamente por razones de implementación. Estos identificadores internos del sistema no deben ser incluidos como atributos.

## Relaciones entre Clases

Existen tres relaciones diferentes entre clases, **Dependencias**, **Generalización y Asociación**. En las relaciones se habla de una clase destino y de una clase origen. El origen es desde la que se realiza la acción de relacionar. Es decir desde la que parte la flecha, el destino es la que recibe la flecha.

En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, es decir especifica cuantas instancias de una clase se pueden relacionar a una sola instancia de otra clase.

Se anotan en cada extremo de la relación y éstas pueden ser:

- **uno-uno** (Fig. 3.6.2)
- **uno-muchos (1...\*)** (Fig. 3.6.3)
- **muchos-muchos(\* \*)** (Fig. 3.6.4)
- **opcional (0..1 )** (Fig. 3.6.4 )
- **número fijo: m** (m denota el número).



*Fig. 3.6.2 Diagrama de clases describiendo una multiplicidad de "uno-uno".*



*Fig. 3.6.3 Diagrama de clases describiendo una multiplicidad de "uno-muchos".*

La notación para representar una relación opcional, donde la multiplicidad es "uno" o "cero", escribiendo una relación opcional, 0 o 1. Esto significa que dos objetos pueden o no estar conectados, y si lo están corresponden a una multiplicidad de 1.



*Fig. 3.6.4 Diagrama de clases describiendo una multiplicidad de "muchos-muchos".*



*Fig. 3.6.5 Diagrama de clases describiendo una multiplicidad "opcional".*

### **Herencia (Especialización/Generalización)**

Las clases con atributos y operaciones comunes se pueden organizar de forma jerárquica, mediante la herencia. La herencia es una abstracción importante para compartir similitudes entre clases, donde todos los atributos y operaciones comunes a varias clases se pueden compartir por medio de la superclase, una clase más general. Las clases más refinadas se conocen como las subclases.

### **Asociación**

La relación entre clases conocida como Asociación, permite asociar objetos que colaboran entre sí. Una asociación describe la relación entre clases de objetos, y describe posibles ligas, donde una liga es una instancia de una asociación, al igual que un objeto es una instancia de una clase.

### **Agregación:**

(El objeto base utiliza al incluido para su funcionamiento). Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye.

### **Composición:**

(El Objeto base se construye a partir del objeto incluido). Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del que lo incluye.

Para nuestro sistema se desarrolló el diagrama de clases con la versión 10.2 de Visual Paradigm for UML Standard Edition (licencia de la BUAP). Fig. 3.6.6

Visual Paradigm for UML Standard Edition (Benemérita Universidad Autónoma de Puebla)

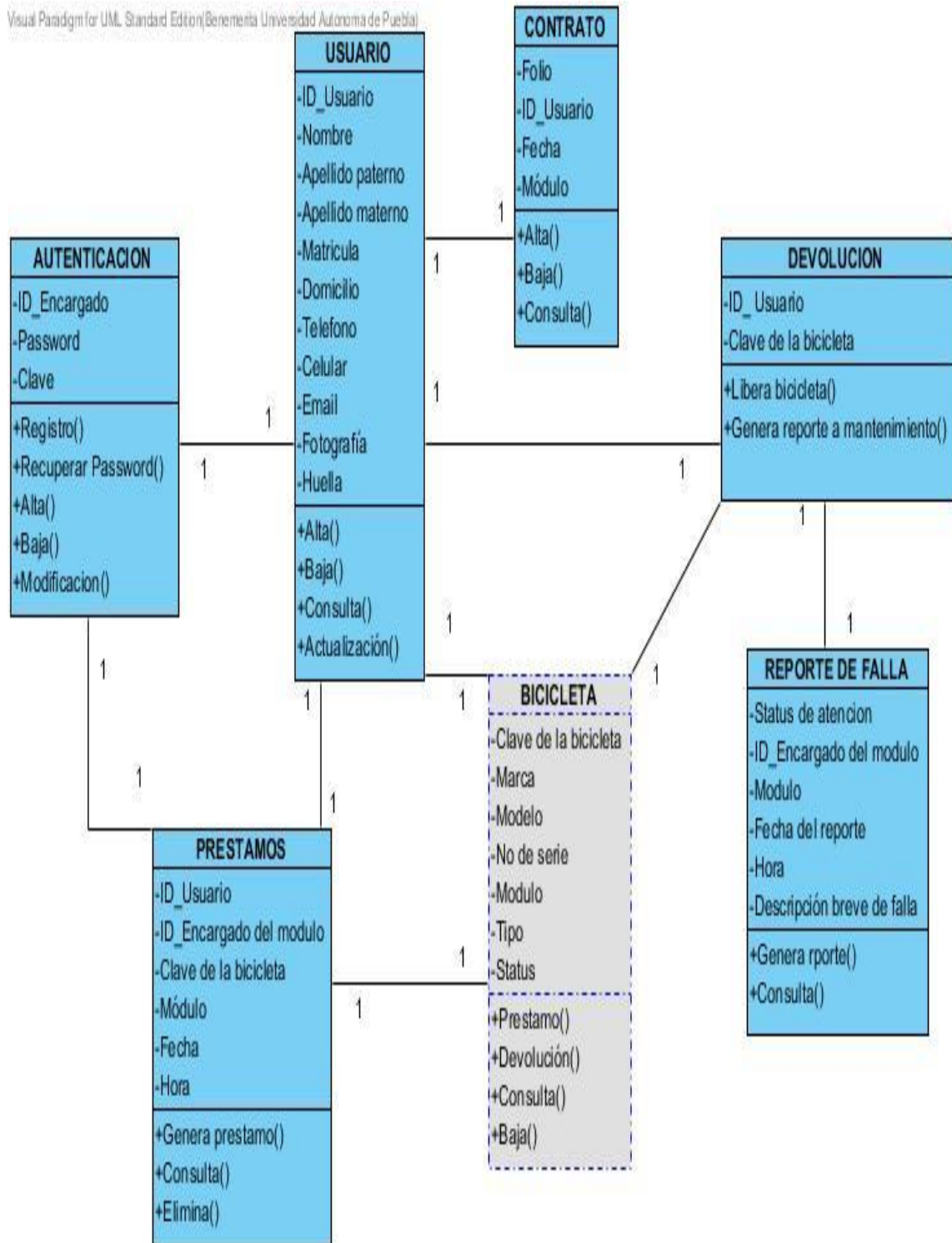


Fig. 3.6.6 Diagrama de clases del Sistema Administración de Bicicletas.

### 3.7 Escenarios

Los escenarios de casos de uso son una descripción del diagrama de casos de uso. Un escenario de caso de uso se divide en tres secciones: identificación e iniciación; pasos realizados y condiciones, suposiciones y preguntas. No hay un formato estandarizado para los casos de uso, por lo que para este sistema “Administración de Bicicletas” especificamos los estándares siguientes. [16]

<b>Sistema:</b> Administración de Bicicletas	
<b>Nombre del Caso de Uso:</b> <i>Autenticación</i>	
<b>Actores:</b> Encargado del Módulo	
<b>Descripción:</b> Permite que el encargado del módulo de su clave para empezar a utilizar la base de datos “Administración de Bicicletas” para proceder al registro de usuarios, préstamo y devolución de bicicletas.	
<b>Evento Desencadenador:</b> El encargado del módulo ya tuvo que haberse dado de alta antes.	
<b>Pasos Realizados (ruta principal)</b>	<b>Información para los pasos</b>
1.- El encargado del módulo ingresada su clave y su password.	La B.D. Verifica si el encargado del módulo ya se registró previamente.
	La B.D da la autenticación del encargado del módulo al dar su password y clave,
<b>Condiciones:</b> El encargado del módulo ya se registró con anterioridad.	
<b>Suposiciones:</b> Autenticación correcta.	

*Tabla 3.7.1 Escenario de autenticación del Sistema Administración de Bicicletas.*



<b>Sistema:</b> Administración de Bicicletas	
<b>Nombre del Caso de Uso:</b> <i>Módulo de Registro</i>	
<b>Actores:</b> Usuario y Encargado del Módulo de Registro	
<b>Descripción:</b> Permite que el usuario se registre mostrando la documentación necesaria.	
<b>Evento Desencadenador:</b> El usuario va al módulo de registro para que lo den de alta, en el Sistema Administración de Bicicletas con identificación oficial vigente.	
<b>Pasos Realizados (ruta principal)</b>	<b>Información para los pasos</b>
1.- El usuario muestra documentación y se identifica con credencial vigente.	Se captura en la B.D. la siguiente documentación: nombre, apellido paterno, apellido materno, matricula, domicilio, teléfono, celular, e-mail, huella digital, fotografía digital, y se le asigna ID.
2.- Verifica si el usuario ya se registró previamente	Búsqueda en la BD por ID del usuario.
3.- El usuario ya existe.	Despliega la BD que el usuario ya existe.
4.- Formato de contrato.	Imprime contrato.
5.- Firma Contrato.	
<b>Condiciones:</b> El usuario tiene documentación en orden.	
<b>Suposiciones:</b> Se registra y se le asigna ID.	

*Tabla 3.7.2 Escenario del módulo de registro del Sistema Administración de Bicicletas.*



<b>Sistema: Administración de Bicicletas</b>	
<b>Nombre del Caso de Uso: Módulo de Préstamo</b>	
<b>Actores:</b> Usuario y Encargado del Módulo de Préstamo.	
<b>Descripción:</b> Permite que el usuario adquiriera una bicicleta a préstamo	
<b>Evento Desencadenador:</b> El usuario se identifica en el módulo con el encargado ID. Para que se le asigne una bicicleta.	
<b>Pasos Realizados (ruta principal)</b>	<b>Información para los pasos</b>
1.- El usuario se identifica con Matricula o Huella Digital.	Revisa existencia en la B.D.
2.- Ingresa la clave de la bicicleta.	Se le asigna bicicleta, se registra préstamo fecha y hora.
<b>Condiciones: Debe haber bicicletas disponibles.</b> El usuario se registró con anterioridad.	
<b>Suposiciones:</b> Se le asigna bicicleta.	

Tabla 3.7.3 Escenario del módulo de préstamo del Sistema Administración de Bicicletas

<b>Sistema:</b> Administración de Bicicletas	
<b>Nombre del Caso de Uso:</b> <i>Módulo de Devolución.</i>	
<b>Actores:</b> Usuario y Encargado del Módulo de Devolución.	
<b>Descripción:</b> Permite la entrega de la bicicleta al encargado del módulo.	
<b>Evento Desencadenador:</b> El usuario se dirige al módulo de devolución para entregar la bicicleta que le fue dada a préstamo.	
<b>Pasos Realizados (ruta principal)</b>	<b>Información para los pasos</b>
1.- El usuario se dirige al módulo de devolución para entregar la bicicleta.	ID revisa a usuario
2.-Se despliega la información de la bicicleta	ID. De bicicleta.
3.- Libera bicicleta.	Abandera registro de disponible.
<b>Condiciones:</b> La bicicleta está en buen estado.	
<b>Suposiciones:</b> La bicicleta está en buen estado	
<b>Preguntas:</b> ¿La bicicleta está en mal estado?	

*Tabla 3.7.4 Escenario del módulo de devolución del Sistema Administración de Bicicletas.*

<b>Sistema:</b> Administración de Bicicletas	
<b>Nombre del Caso de Uso:</b> <i>Módulo de Devolución. "BICICLETA EN MAL ESTADO"</i>	
<b>Actores:</b> Usuario y Encargado del Módulo de Devolución.	
<b>Descripción:</b> Permite la entrega de la bicicleta al encargado del módulo.	
<b>Evento Desencadenador:</b> El usuario se dirige al módulo de devolución para entregar la bicicleta que le fue dada a préstamo.	
<b>Pasos Realizados (ruta principal)</b>	<b>Información para los pasos</b>
1.- El usuario se dirige al módulo de devolución para entregar la bicicleta.	ID revisa a usuario
2.-Se despliega la información de la bicicleta	ID. De bicicleta.
3.- El encargado del módulo revisa bicicleta.	La bicicleta está en mal estado.
4.- Genera reporte de Falla.	Se envía a mantenimiento.
<b>Condiciones:</b> La bicicleta está en buen estado.	
<b>Suposiciones:</b> La bicicleta está en mal estado	
<b>Preguntas:</b> N/A	

*Tabla 3.7.5 Escenario del módulo de devolución "bicicleta en mal estado"*

*Del Sistema Administración de Bicicletas.*



### **3.8 Diagrama de Secuencia**

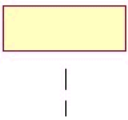


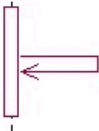
Este diagrama (también llamado diagrama de interacción) muestra las interacciones entre un conjunto de objetos (clases, actores), ordenadas según el tiempo en que tienen lugar. Es decir, muestra el orden de las llamadas en el sistema. Se utiliza un diagrama para cada llamada a representar. Es imposible representar en un solo diagrama la secuencia de todas las llamadas posibles del sistema, es por ello que se escoge un punto de partida. El diagrama se compone con los objetos que forman parte de la secuencia, estos se sitúan en la parte superior de la pantalla, normalmente a la izquierda se sitúa el que inicia la acción. De estos objetos sale una línea que indica su vida en el sistema. Esta línea simple se convierte en una línea gruesa cuando representa que el objeto tiene el foco del sistema, es decir cuando él está activo.

El objeto puede existir sólo durante la ejecución de la interacción, se puede crear o puede ser destruido durante la ejecución de la interacción.

En este tipo de diagramas también intervienen los mensajes, que son la forma en que se comunican los objetos: el objeto origen solicita (llama a) una operación del objeto destino. El diagrama de secuencia puede ser obtenido de dos partes, desde el Diagrama Estático de Clases o desde el de Casos de Usos. [16]

## Elementos

Los componentes de un diagrama de interacción se muestran en la siguiente figura (Fig. 3.8.1).

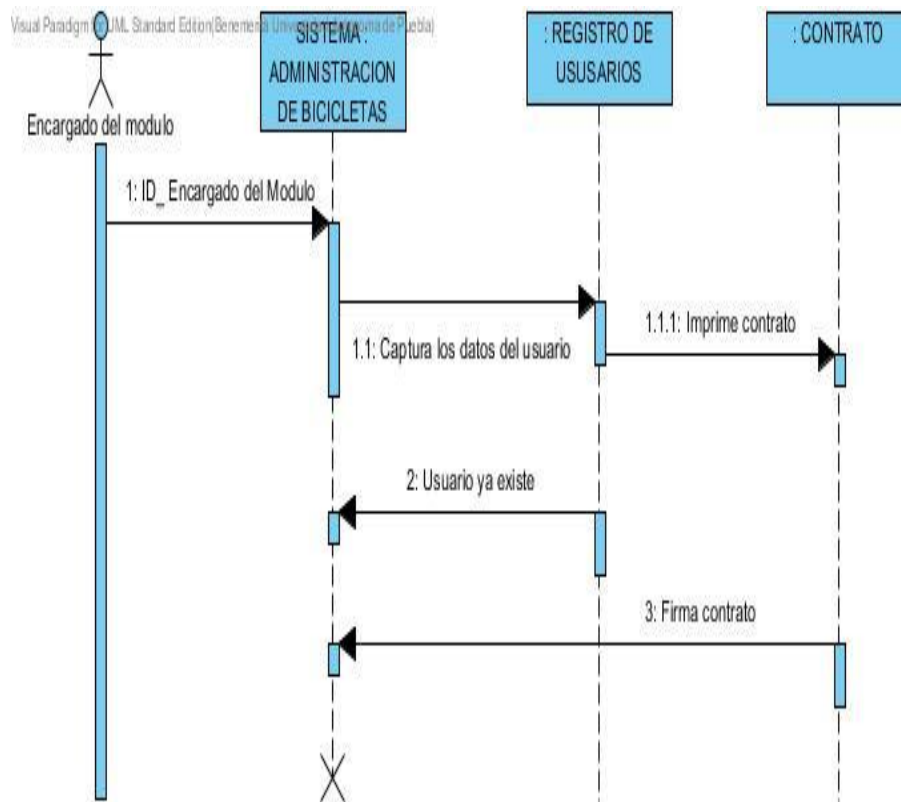
Nombre	Símbolo	Significado
<b>Línea de vida</b>		Un objeto (o actor) se representa como una línea vertical punteada con un rectángulo de encabezado y con rectángulos a través de la línea principal que denotan la ejecución de métodos (activación). El rectángulo de encabezado contiene el nombre del objeto y el de su clase.
<b>Activación.</b>		Muestra el periodo de tiempo en el cual el objeto se encuentra desarrollando alguna operación, bien sea por sí mismo o por medio de delegación a alguno de sus atributos. Se denota como un rectángulo delgado sobre la línea de vida del objeto.
<b>Mensajes</b>		El envío de mensajes entre objetos se denota mediante una línea sólida dirigida, desde el objeto que emite el mensaje hacia el objeto que lo ejecuta. Representa la llamada de un método (operación) de un objeto en particular.
		También es posible visualizar llamadas a métodos desde el mismo objeto en estudio.
	X	Una X en la parte inferior de la línea de vida representa el momento en que se destruye el objeto

*Fig. 3.8.1 Simbología del diagrama de secuencia*

Los diagramas de secuencia que se desarrollaron para nuestro sistema son los siguientes:

➤ **Diagrama de secuencia registro de usuarios (Fig. 3.8.2).**

El encargado del módulo captura los datos del nuevo usuario, verifica que no exista en la base de datos imprime el contrato y el usuario firma el contrato.



*Fig. 3.8.2 Diagrama de secuencia registro de usuarios del Sistema Administración de Bicicletas*

➤ Diagrama de secuencia préstamo de bicicletas (Fig. 3.8.3).

El encargado del módulo inserta el id\_usuario y la clave de la bicicleta para registrar el préstamo.

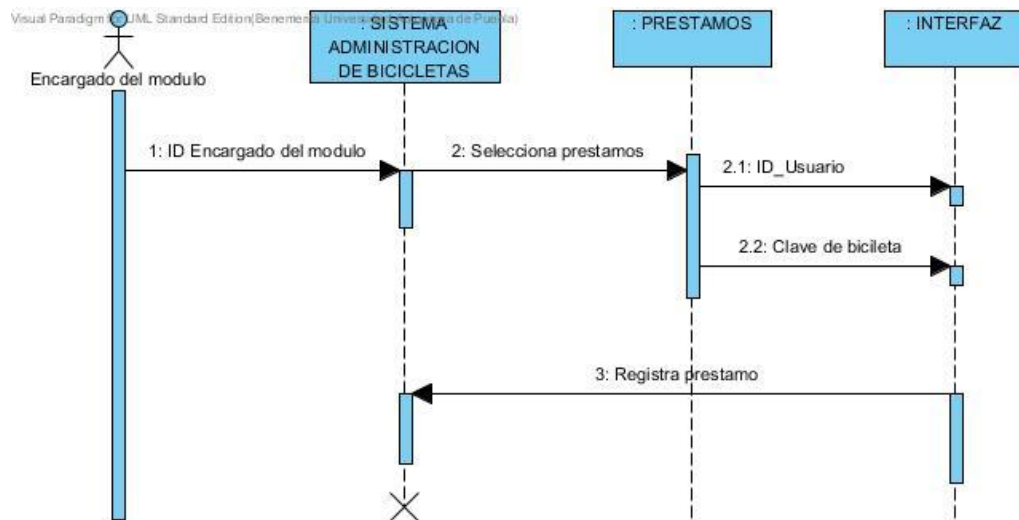


Fig. 3.8.3 Diagrama de secuencia préstamo del Sistema Administración de Bicicletas

➤ Diagrama de secuencia devolución de bicicletas (Fig. 3.8.4)

El encargado del módulo inserta el id\_usuario y la clave de la bicicleta para registrar la devolución, si presenta algún daño la bicicleta se genera un reporte de falla a mantenimiento.

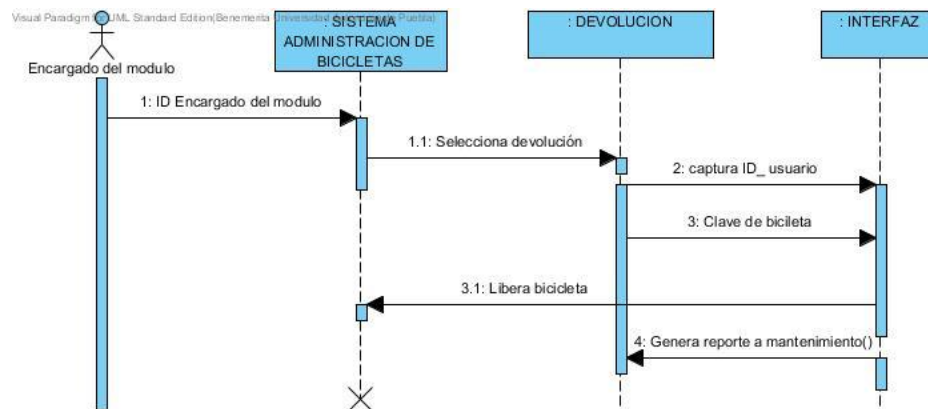


Fig. 3.8.4 Diagrama de secuencia devolución del Sistema Administración de Bicicletas.

## 4. DISEÑO DE LA BASE DE DATOS

---

Una vez hecho el levantamiento de requerimientos y el análisis del sistema, es importante plasmar esta información en el diseño de la base de datos, la cual se realizó a través del modelo conceptual, con el diagrama entidad - relación, la definición de los elementos que intervienen como son entidades, atributos, dominios y relaciones. [18]

### 4.1 Modelo Entidad - Relación

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas. La simbología que distingue a un diagrama de este tipo está mostrada en la Figura 4.1.1.

Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y las jerarquías de generalización, en lo que se ha denominado modelo entidad-relación extendido.

**Entidad** Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo: coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.



Hay dos tipos de entidades: *fuertes y débiles*. Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad. Una entidad fuerte es una entidad que no es débil.

**Relación (interrelación)** Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior.

Las entidades que están involucradas en una determinada relación se denominan entidades participantes. El número de participantes en una relación es lo que se denomina grado de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación binaria; si son tres las entidades participantes, la relación es ternaria; etc.

Una relación recursiva es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles. El nombre de estos papeles es importante para determinar la función de cada participación.

La cardinalidad con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. La participación de una entidad en una relación es obligatoria (total) si la existencia de cada una de sus ocurrencias requiere la existencia de, al menos, una ocurrencia de la otra entidad participante. Si no, la participación es opcional (parcial). Las reglas que definen la cardinalidad de las relaciones son las reglas de negocio.

**Atributo** Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante bolitas que cuelgan de las entidades o relaciones a las que pertenecen.



Cada atributo tiene un conjunto de valores asociados denominado dominio. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio.

Los atributos pueden ser simples o compuestos. Un atributo simple es un atributo que tiene un solo componente, que no se puede dividir en partes más pequeñas que tengan un significado propio. Un atributo compuesto es un atributo con varios componentes, cada uno con un significado por sí mismo. Un grupo de atributos se representa mediante un atributo compuesto cuando tienen afinidad en cuanto a su significado, o en cuanto a su uso. Un atributo compuesto se representa gráficamente mediante un óvalo.

Los atributos también pueden clasificarse en monovalentes o polivalentes. Un atributo monovalente es aquel que tiene un solo valor para cada ocurrencia de la entidad o relación a la que pertenece. Un atributo polivalente es aquel que tiene varios valores para cada ocurrencia de la entidad o relación a la que pertenece. A estos atributos también se les denomina multivaluados, y pueden tener un número máximo y un número mínimo de valores.

La cardinalidad de un atributo indica el número mínimo y el número máximo de valores que puede tomar para cada ocurrencia de la entidad o relación a la que pertenece. El valor por omisión es (1, 1).

Por último, los atributos pueden ser derivados. Un atributo derivado es aquel que representa un valor que se puede obtener a partir del valor de uno o varios atributos, que no necesariamente deben pertenecer a la misma entidad o relación.

**Identificador** Un identificador de una entidad es un atributo o conjunto de atributos que determina de modo único cada ocurrencia de esa entidad. Un identificador de una entidad debe cumplir dos condiciones:



1. No pueden existir dos ocurrencias de la entidad con el mismo valor del identificador.
2. Si se omite cualquier atributo del identificador, la condición anterior deja de cumplirse.

Toda entidad tiene al menos un identificador y puede tener varios identificadores alternativos.

**Jerarquía de generalización** Una entidad E es una generalización de un grupo de entidades E1, E2, ... En, si cada ocurrencia de cada una de esas entidades es también una ocurrencia de E. Todas las propiedades de la entidad genérica E son heredadas por las subentidades.

Cada jerarquía es total o parcial, y exclusiva o superpuesta. Una jerarquía es total si cada ocurrencia de la entidad genérica corresponde al menos con una ocurrencia de alguna subentidad. Es parcial si existe alguna ocurrencia de la entidad genérica que no corresponde con ninguna ocurrencia de ninguna subentidad. Una jerarquía es exclusiva si cada ocurrencia de la entidad genérica corresponde, como mucho, con una ocurrencia de una sola de las subentidades. Es superpuesta si existe alguna ocurrencia de la entidad genérica que corresponde a ocurrencias de dos o más subentidades diferentes. Un subconjunto es un caso particular de generalización con una sola entidad como subentidad. Un subconjunto siempre es una jerarquía parcial y exclusiva.

### Elementos de un Diagrama de Entidad – Relación

Nombre	Símbolo
Entidad	
Entidad Débil	
INTERRELACIÓN	
INTERRELACION DEBIL	
Atributo	
Atributo Llave	
Atributo Derivado	
Atributo Compuesto	

*Fig. 4.1.1 Simbología diagrama entidad relación*

## 4.2 Diagrama Entidad - Relación

El diagrama Entidad – Relación que describe nuestro sistema se muestra en la Fig. 4.2.1

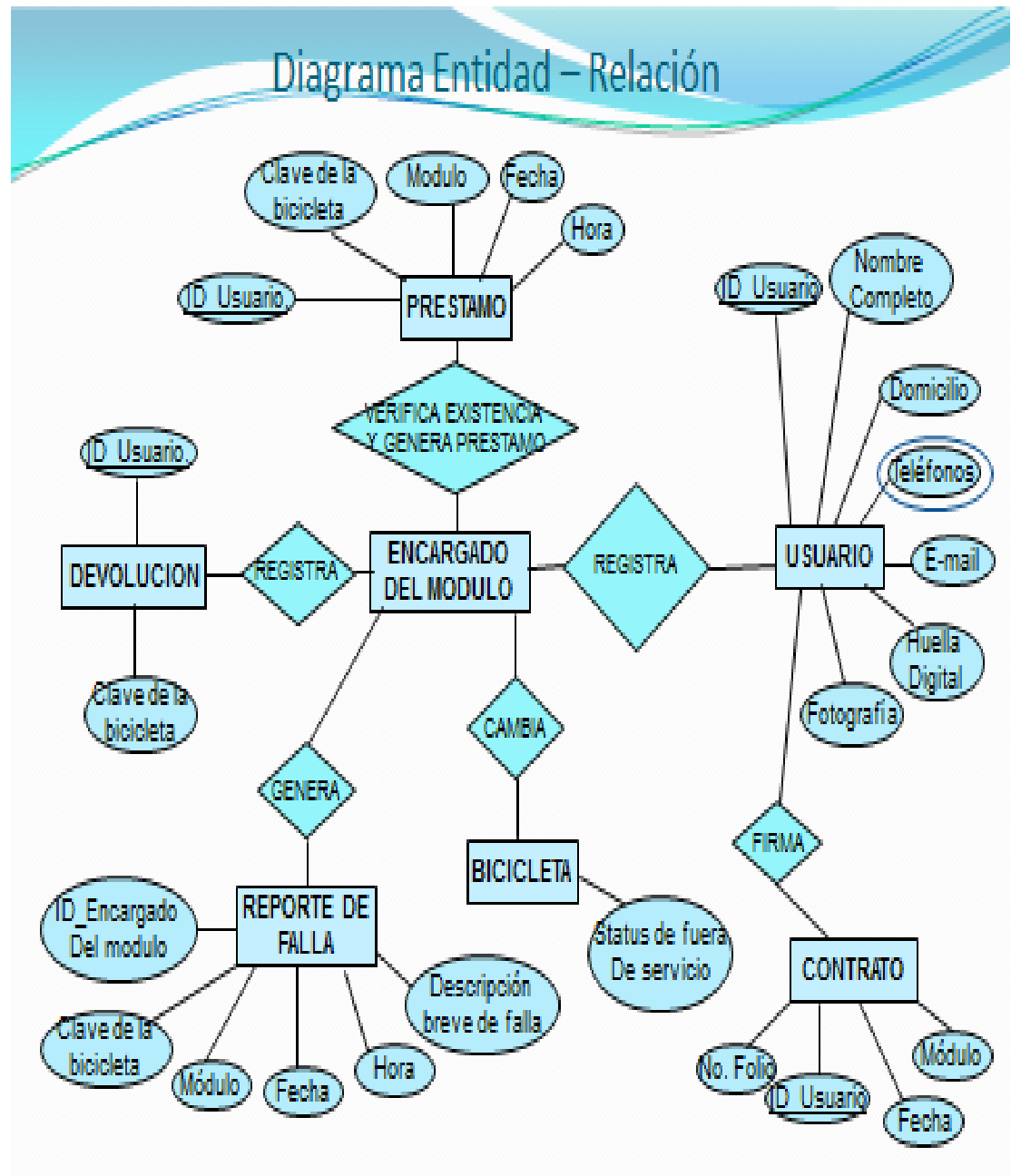


Fig. 4.2.1 Diagrama entidad relación del Sistema Administración de Bicicletas

### 4.3 Base de Datos

Siguiendo el modelo entidad relación que comprende el sistema se desarrollan las tablas con sus campos correspondientes como se muestran en las siguientes figuras.

- ✓ Fig. 4.3.1 Base de Datos del Sistema
- ✓ Fig. 4.3.2 Tabla *bici\_administradores*
- ✓ Fig. 4.3.3 Tabla *bici\_usuarios*
- ✓ Fig. 4.3.4 Tabla *bici\_presdev*
- ✓ Fig. 4.3.5 Tabla *contrato\_usu*
- ✓ Fig. 4.3.5 Tabla *modulo*
- ✓ Fig. 4.3.5 Tabla *rep\_falla*

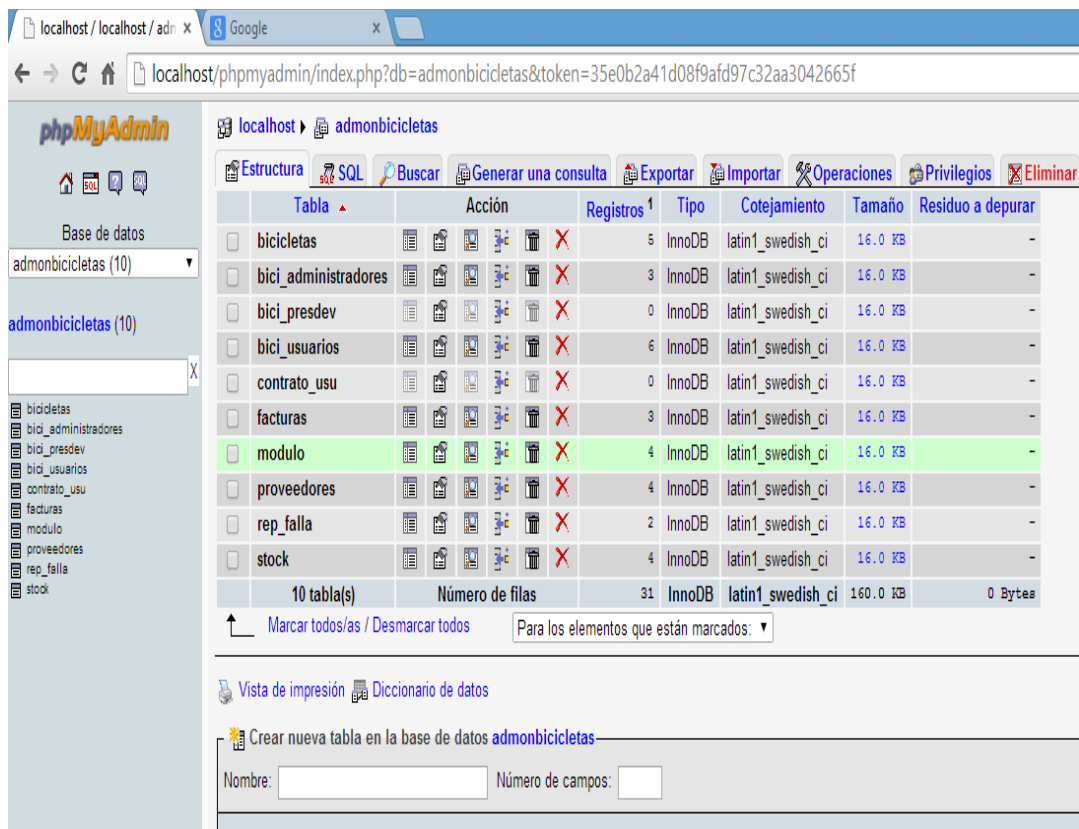


Fig. 4.3.1 Base de Datos del Sistema Administración de Bicicletas.

id_admin	nombre_admin	apaterno_admin	amaterno_admin	domicilio_admin	telefono1_admin	telefono2_admin	email_admin	huella
lourdes	maria de lourdes	montiel	gonzalez	nuestra señora del rosario 13 c la guadalupana	2221293824	2221716496	manyly_mon@yahoo.com.mx	NULL
marco	marco antonio	espinosa	garcia	guadalupana puebla	2221293824	2225083429	maeg72@hotmail.com	NULL
seve_espi	Severiano	Espinosa	Muñoz	COI. Zaragoza	2357454	2221608964	s_espi@yahoo.com	NULL

[Todos/as](#) / [Desmarcar todos](#) Para los elementos que están marcados:

Mostrar:  filas empezando de

horizontal y repetir los encabezados cada  celdas

[Previsualización para imprimir \(documento completo\)](#) [Exportar](#) [CREATE VIEW](#)

Fig. 4.3.2 Tabla bici\_administradores del Sistema Administración de Bicicleta.

localhost > admonbicicletas > bici\_usuarios

[Examinar](#) [Estructura](#) [SQL](#) [Buscar](#) [Insertar](#) [Exportar](#) [Importar](#) [Operaciones](#) [Vaciar](#) [Eliminar](#)

Mostrando registros 0 - 5 (~6<sup>1</sup> total, La consulta tardó 0.0006 seg)

```

SELECT *
FROM 'bici_usuarios'
LIMIT 0 , 30
  
```

Perfil/Perfilamiento [ [Editar](#) ] [ [Explicar el SQL](#) ] [ [Crear código PHP](#) ] [

Mostrar:  filas empezando de

en modo horizontal y repetir los encabezados cada  celdas

Organizar según la clave: Ninguna

+ Opciones

	id_usuario	nombre_usu	apaterno_usu	amaterno_usu	domicilio_usu	telefono1_usu	telefono2_usu	email_usu
<input type="checkbox"/>	880611173	marco antonio	espinosa	montiel	la guadalupana	1293824	2225083429	bross619@hotmail.com
<input type="checkbox"/>	880611175	Severiano	Espinosa	Muñoz	5 de febrero No. 11. Zaragoza	2357454	2221608964	seve@yahoo.com
<input type="checkbox"/>	90234156896	Mario	Montiel	Gonzalez	Santa Cruz Tlaxcala	24611223456	012464612654	mario@hotmail.com
<input type="checkbox"/>	903245678	Paty	Montiel	Rocha	rio papaloapan 6107 colonia	2357454	223134567	paty_rocha@yahoo.com.mx
<input type="checkbox"/>	90999999999	JESUS	MONIEL	RIVERA	AVENIDA PINOS 9014	2414185000	2414133333	mario-montiel-gonzalez@hotmail
<input type="checkbox"/>	912210813	Rocio	Cortes	Morales	Bld. Valsequillo 1423	2334455	000000	chio_cor@yahoo.com

[Marcar todos/as](#) / [Desmarcar todos](#) Para los elementos que están marcados:

Mostrar:  filas empezando de

Fig. 4.3.3 Tabla bici\_usuarios del Sistema Administración de Bicicleta.

MySQL ha devuelto un valor vacío (i.e., cero columnas). (La consulta tardó 0.0004 seg)

```
SELECT *
FROM `bici_presdev`
LIMIT 0 , 30
```

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	id_usuario	char(20)	latin1_swedish_ci		Sí	NULL		
<input type="checkbox"/>	num_serie	char(20)	latin1_swedish_ci		Sí	NULL		
<input type="checkbox"/>	modulo	char(1)	latin1_swedish_ci		Sí	NULL		
<input type="checkbox"/>	fecha_pres	date			Sí	NULL		
<input type="checkbox"/>	hora_pres	time			Sí	NULL		
<input type="checkbox"/>	status_bici	char(1)	latin1_swedish_ci		Sí	NULL		
<input type="checkbox"/>	fecha_dev	date			Sí	NULL		
<input type="checkbox"/>	hora_dev	time			Sí	NULL		

Vista de impresión  
  Vista de relaciones  
  Planteamiento de la estructura de tabla

Añadir  campo(s)  
  Al final de la tabla  
  Al comienzo de la tabla  
  Después de   
 Continuar

Fig. 4.3.4 Tabla bici\_presdev del Sistema Administración de Bicicletas.

MySQL ha devuelto un valor vacío (i.e., cero columnas). (La consulta tardó 0.0005 seg)

```
SELECT *
FROM `contrato_usu`
LIMIT 0 , 30
```

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	folio	int(11)			No	None	AUTO_INCREMENT	
<input type="checkbox"/>	id_usuario	char(20)	latin1_swedish_ci		Sí	NULL		
<input type="checkbox"/>	modulo	char(1)	latin1_swedish_ci		Sí	NULL		
<input type="checkbox"/>	fecha_cont	date			Sí	NULL		

Vista de impresión  
  Vista de relaciones  
  Planteamiento de la estructura de tabla

Añadir  campo(s)  
  Al final de la tabla  
  Al comienzo de la tabla  
  Después de   
 Continuar

Índices: @

Acción	Nombre de la clave	Tipo	Único	Empacado	Campo	Cardinalidad	Cotejamiento	Nulo	Comentario
	PRIMARY	BTREE	Sí	No	folio	0	A		

Fig. 4.3.5 Tabla contrato\_usu del Sistema Administración de Bicicletas.

localhost ▶ admonbicicletas ▶ modulo

Mostrando registros 0 - 3 (~4<sup>1</sup> total, La consulta tardó 0.0005 seg)

```
SELECT *
FROM 'modulo'
LIMIT 0 , 30
```

Mostrar : 30 filas empezando de 0  
en modo horizontal y repetir los encabezados cada 100 celdas

Organizar según la clave: Ninguna

+ Opciones

	mod_clave	mod_direccion
<input type="checkbox"/>	1	arquitectura
<input type="checkbox"/>	2	polideportivo
<input type="checkbox"/>	3	contaduria
<input type="checkbox"/>	4	computacion

Mostrar : 30 filas empezando de 0  
en modo horizontal y repetir los encabezados cada 100 celdas

Fig. 4.3.6 Tabla modulo del Sistema Administración de Bicicletas.

localhost ▶ admonbicicletas ▶ rep\_falla

Mostrando registros 0 - 1 (~2<sup>1</sup> total, La consulta tardó 0.0008 seg)

```
SELECT *
FROM 'rep_falla'
LIMIT 0 , 30
```

Mostrar : 30 filas empezando de 0  
en modo horizontal y repetir los encabezados cada 100 celdas

Organizar según la clave: Ninguna

+ Opciones

	salida_prob	id_admin	num_serie	descripcion_falla	mod_clave	fecha_rep	hora_rep	num_reporte
<input type="checkbox"/>	2014-05-29	lourdes	b0001	llanta trasera ponchada	2	2014-05-22	18:30:00	2
<input type="checkbox"/>	2014-06-12	lourdes	b0001	llanta trasera ponchada	2	2014-06-02	13:45:00	3

Fig. 4.3.6 Tabla rep\_falla del Sistema Administración de Bicicletas.



## 5. IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA

---

En el presente capítulo se muestra la implementación y pruebas del funcionamiento de la Aplicación Web. Tomando como base el diseño de la base de datos, se ha generado el código para la creación de la base de datos (Diseño físico). Las herramientas que se utilizaron para lograr dicho objetivo, se mencionan en las siguientes secciones. Además se muestran las pruebas que se realizaron a la Aplicación.

### 5.1 Implementación

Para la implementación de la Aplicación, se utilizó el lenguaje HTML, el cual de manera tradicional se editó en block de notas, así como también nos apoyamos en el programa Dreamweaver. Para la programación de la Aplicación Web, se ha utilizado el lenguaje PHP.

Se establece como punto de publicación, un servidor Web Apache que nos brinda un servicio potente, flexible, modular y que además nos permite optimizar los recursos del servidor y obtener mejor resultados en los tiempos de procesamiento y respuesta de las peticiones. Por otra parte es confiable ya que se integra de forma natural con PHP.

Se usó como manejador de base de datos MySQL y PhpMyAdmin para la administración de la misma. Las herramientas anteriormente mencionadas funcionan en conjunto y están integradas en un servidor independiente llamado WampServer, el cual fue el que se utilizó para el desarrollo de esta Aplicación, que consiste principalmente en la base de datos MySql, el servidor Web Apache y los intérpretes para lenguaje de script PHP y Javascript.

## 5.2 Implementación de la Interfaz

### Página Inicial

Al ingresar a la Aplicación Web desde cualquier navegador, lo primero que se presenta es la interfaz de la página de inicio (Figura 5.2.1), en ella se puede apreciar un texto que describe el nombre del sistema así como su slogan, y los botones para que se identifique y el recaptcha.



*Fig. 5.2.1 Página de inicio*

Una vez que se ha identificado correctamente entra a una segunda pantalla donde hay cinco ligas, con los cual el usuario puede navegar dentro de la Aplicación. (Fig. 5.2.2)



Fig. 5.2.2 Préstamo y Devolución

### Alta/Baja de usuarios

El botón Alta/Baja de usuarios, muestra un formulario a llenar que le permite al encargado del módulo registrar al usuario (Figura 5.2.3) y poder tener el beneficio de adquirir una bicicleta a préstamo.

Fig. 5.2.3. Alta de usuarios

## Modificar

Al utilizar el botón actualizar, se selecciona el usuario el cual se desea modificar; después de esto se muestra un formulario donde se permite realizar dicha acción. (Fig. 5.2.4)

ID_USUARIO	NOMBRE	A_PATERNO	A_MATERNO	DOMICILIO	TELEFONO1	TELEFONO2	E-MAIL	ACTUALIZAR
880611173	marco antonio	espinosa	montiel	la guadalupana	1293824	2225083429	bross619@hotmail.com	
880611175	Severiano	Espinosa	Muñoz	5 de febrero No. 11. Zaragoza	2357454	2221608964	seve@yahoo.com	
90234156896	Mario	Montiel	Gonzalez	Santa Cruz Tlaxcala	24611223456	012464612654	mario@hotmail.com	
903245678	Paty	Montiel	Rocha	rio papaloapan 6107 colonia	2357454	223134567	paty_rocha@yahoo.com.mx	
				AVENIDA			mario-montiel-	

Fig. 5.2.4. Modificar usuarios

## Baja de usuarios

Se selecciona el usuario el cual se desea dar de baja y aparece una leyenda que dice que si realmente se quiere eliminar el usuario. (Fig. 5.2.5)

**citius bike » dar de baja a usuario**

Regresar

Eliminación de Usuario de la base de Datos

ID	NOMBRE	PATERNO	MATERNO	DOMICILIO	TELEFONO1	TELEFONO2	EMAIL	ELIMINAR
880611173	marco antonio	espínosa	montiel	la guadalupana	1293824	2225083429	bross619@hotmail.com	✖
880611175	Severiano	Espinosa	Muñoz	5 de febrero No. 11. Zaragoza	2357454	2221608964	seve@yahoo.com	✖
90234156896	Mario	Montiel	Gonzalez	Santa Cruz Tlaxcala	24611223456	012464612654	mario@hotmail.com	✖
903245678	Paty	Montiel	Rocha	rio papaloapan 6107 colonia	2357454	223134567	paty_rocha@yahoo.com.mx	✖
90999999999	JESUS	MONIEL	RIVERA	AVENIDA PINOS 9014	2414185000	2414133333	mario-montiel-gonzalez@hotmail	✖
912210813	Rocio	Cortes	Morales	Bld. Valsequillo 1423	2334455	000000	chio_cor@yahoo.com	✖

Fig. 5.2.5. Eliminar Usuario

### Consulta usuario

Muestra la información del usuario que se desea consultar. (Fig. 5.2.6)

**citius bike » consultas**

Regresar

**consulta usuario**

Datos de los usuarios

ID	NOMBRE	AP PATERNO	AP MATERNO	DOMICILIO	TELEFONO1	TELEFONO2	EMAIL
880611173	marco antonio	espínosa	montiel	la guadalupana	1293824	2225083429	bross619@hotmail.com
880611175	Severiano	Espinosa	Muñoz	5 de febrero No. 11. Zaragoza	2357454	2221608964	seve@yahoo.com

Fig. 5.2.6. Consulta usuario

## Préstamo de Bicicletas

En esta pantalla se le pide al usuario su clave y se le asigna la bicicleta.

(Fig. 5.2.7)

citius bike » préstamo de bicicletas

Devolucion | Regresar

ingresa la clave del usuario y de la bicicleta.

No. de Serie de bicicleta:

Id Usuario :

Grabar

Fig. 5.2.7 Préstamo de bicicletas

## Devolución de Bicicletas

En esta pantalla se pone el número de serie de la bicicleta para que se marque como devuelta. (Fig. 5.2.8.)

citius bike » devolución de bicicletas

Regresar

ingresa los datos de la bicicleta

No. de Serie :

Grabar

Fig. 5.2.8 Devolución de bicicletas

## Reporte de falla

En esta pantalla se ingresan los datos que se muestran para generar el reporte.  
(Fig. 5.2.9.)

The screenshot shows a web interface for generating a report. At the top, it says 'citius bike » genera reporte de falla'. Below this is a navigation bar with 'Consulta Reportes' and 'Regresar'. The main heading is 'ingresa los datos para generar el reporte'. The form contains the following fields:

- Id\_administrador:
- Módulo:
- Fecha:
- Hora:
- Id bicicleta:
- Fecha de salida probable :
- Descripción de falla :

At the bottom left, there is a 'Grabar' button.

Fig. 5.2.8 Devolución de bicicletas



## 6. CONCLUSIONES

---

La Aplicación Web se desarrolló e implementó satisfactoriamente para el Sistema, “Administración de Bicicletas” en el módulo de “Registro de Usuarios y Préstamo” dentro de un campus.

Se utilizaron diferentes programas, pero la base más importante fue a través de WampServer, que proporciona, PhpMyadmin, Php, Apache y MySQL, siendo muy confiables en el desarrollo de la Aplicación.

Producto del análisis y diseño basado en la metodología de desarrollo en espiral, se redujeron tiempos en la programación de la aplicación y en la estructura de datos y usuarios que integran la aplicación.

Se optimizaron datos y condiciones que son necesarias para el sistema al utilizar el modelo relacional, y se evitó la redundancia de datos, las anomalías de actualización y eliminación al normalizar la Base de Datos.

Las pruebas realizadas fueron satisfactorias, éstas se realizaron usando la Prueba Basada en Requisitos o Pruebas de Casos de Uso. De esta manera fue posible comprobar que la Aplicación Web funcionó al 100% con los requerimientos que fueron establecidos.



## 7. TRABAJOS A FUTURO

---

Después de que los objetivos planteados para el desarrollo de la Aplicación Web para este trabajo de Tesis se cumplieron satisfactoriamente, se plantean algunos trabajos a futuro para mejorar la Aplicación, a continuación se mencionan algunos de los más significativos:

- ✓ Integrar un chip en cada bicicleta para tener un mejor control de su ubicación.
- ✓ Implementar la tarjeta electrónica (candados electromagnéticos).
- ✓ Implementar el módulo de seguridad.
- ✓ Digitalizar la fotografía de usuario y huella digital.



## 8. Bibliografía

---

- [1] «QuimiNet.com,» Luz Azul , 21 Julio 2011. [En línea]. Available: <http://www.quiminet.com/articulos/la-bicicleta-como-medio-de-transporte-las-ventajas-de-usar-la-bicicleta-como-medio-de-transporte-2555670.htm>. [Último acceso: 03 Noviembre 2013].
- [2] «Excmo. Ayuntamiento de Alhama de Murcia 2008,» 2008. [En línea]. Available: [http://www.alhamademurcia.es/medio\\_ambiente/ventajas\\_usar\\_bicicleta\\_transporte\\_alternativ.asp](http://www.alhamademurcia.es/medio_ambiente/ventajas_usar_bicicleta_transporte_alternativ.asp). [Último acceso: 03 Noviembre 2013].
- [3] C. J. Date, Introducción a los Sistemas de Bases de Datos, Pearson Addison Wesley, 2001..
- [4] Nuevas Tecnologías de Información y Comunicación, Noviembre 2008. [En línea]. Available: <http://rycsa.files.wordpress.com/2008/12/dfd-teoria.doc>. [Último acceso: 01 Diciembre 2013].
- [5] «Programación Orientada a Objetos con Java,» Facultad de Ingeniería UNAM, [En línea]. Available: <http://profesores.fi-b.unam.mx/carlos/toc.html>. [Último acceso: 05 Diciembre 2013].
- [6] Juan Ramón López Rodríguez, «Licenciatura en Documentación: Bases de datos documentais,» [En línea]. Available: <http://docencia.lbd.udc.es/bdd/teoria/tema1/1.1-IntroduccionALasBasesDeDatos.pdf>. [Último acceso: 20 Diciembre 2013].
- [7] «Normalización de Bases de Datos,» [En línea]. Available: <http://www.buenastareas.com/ensayos/Normalizaci%C3%B3n-De-Bases-De-Datos/1042202.html> [. [Último acceso: 18 Diciembre 2013].
- [8] P. C. Ángel Cobo, PHP y MySQL Tecnologías para el desarrollo de aplicaciones Web, Ed. Díaz de Santos, 2005..
- [9] J. L. Quijado, Domine PHP y MySQL, programación dinámica en el lado de servidor, Alfaomega, RA-MA, 2007.
- [10] J. M. Varese, Dreamweaver Primeros Pasos, Macromedia Inc.,, 2005.



- [11] C. M. G. Segura, Programación Web con PHP y MySQL, Notas, Yucatán: Tizmín, 2008.
- [12] Initiation a Photoshop CS2, Hewlett-Packard Development Company, L.P., 2007.
- [13] «Wikipedia,Ciclovia,» [En línea]. Available: <http://es.wikipedia.org/wiki/Ciclov%C3%ADa>. [Último acceso: 06 Noviembre 2013].
- [14] «Apatzingán Michoacán Diciembre 2010,» INSTITUTO TECNOLÓGICO SUPERIOR DE APATZINGÁN, 2010 Diciembre. [En línea]. Available: [http://es.wikipedia.org/wiki/Metodolog%C3%ADa\\_de\\_desarrollo\\_de\\_software](http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software). [Último acceso: 15 Diciembre 2013].
- [15] «Grupo Espiral,» 1 Agosto 2009. [En línea]. Available: <http://es.geocities.com/modeloespiral/definicion.htm>. [Último acceso: 16 Diciembre 2013].
- [16] J. E. K. KENNETH E. KENDALL, Análisis y Diseño de Sistemas, Octava Edición, México: Pearson, 2011.
- [17] «Tutorial de UML (Modelo de Clases),» [En línea]. Available: <http://users.dcc.uchile.cl/~psalinas/uml/modelo.html>. [Último acceso: 2013 Diciembre].
- [18] D. M. Kroenke, «Procesamiento de Bases de Datos "Fundamentos Diseño e Implementación",» [En línea]. Available: <http://u-libros.com/?reqp=1&reqr=>. [Último acceso: 2014 Enero].
- [19] Marco Antonio Espinosa García. ADMINISTRACIÓN DE BICICLETAS: MÓDULO DE MANTENIMIENTO. Tesis para obtener el grado de Licenciado en Ciencias de la Computación. FCC-BUAP. Junio de 2014

## 9. APÉNDICE A

---

### Creación de la base de datos y las tablas

Se crearon desde PhpMyAdmin:

Primeramente se creó la base de datos, cuyo nombre admonbicicletas. Una vez creada, se crearon las tablas con las que cuenta.

```
-----TABLA DE ADMINISTRADORES-----
create table bici_administradores(
id_admin char(20) not null primary key,
nombre_admin char(30),
apaterno_admin char(15),
amaterno_admin char(15),
domicilio_admin char(60),
telefono1_admin char(12),
telefono2_admin char(12),
email_admin char(40),
huelladig_admin char(20),
fotografia_admin char(20)
);

-----TABLA CONTRATO-----
create table contrato_usu(
folio int not null auto_increment primary key,
id_usuario char(20),
fecha_cont date,
modulo char(1)
);

-----TABLA DE MOVIMIENTOS PRESTAMO Y DEVOLUCION-----

create table bici_presdev(
id_usuario char(20),
num_serie char(20),
modulo char(1),
fecha_pres date,
hora_pres time,
status_bici char(1),
fecha_dev date,
hora_dev time
);

mysql> use admonbicicletas;
Database changed
mysql> show tables;
+-----+
| Tables_in_admonbicicletas |
+-----+
| bici_administradores     |
| bici_presdev              |
| bici_usuarios             |
| bicicletas                |
| contrato_usu              |
| facturas                  |
+-----+
```



```
| contrato_usu  
| facturas  
| proveedores  
| rep_falla  
+-----+  
8 rows in set (0.00 sec)  
  
mysql> alter table contrato_usu  
-> drop column fecha;  
Query OK, 0 rows affected (0.19 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> describe contrato_usu;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type      | Null | Key | Default | Extra      |  
+-----+-----+-----+-----+-----+-----+  
| folio      | int(11)   | NO   | PRI | NULL    | auto_increment  
| id_usuario | char(20)  | YES  |     | NULL    |  
| modulo     | char(1)   | YES  |     | NULL    |  
+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)  
  
mysql> alter table contrato_usu  
-> add column fecha_cont date;  
Query OK, 0 rows affected (0.48 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> describe contrato_usu;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type      | Null | Key | Default | Extra      |  
+-----+-----+-----+-----+-----+-----+  
| folio      | int(11)   | NO   | PRI | NULL    | auto_increment  
| id_usuario | char(20)  | YES  |     | NULL    |  
| modulo     | char(1)   | YES  |     | NULL    |  
| fecha_cont | date      | YES  |     | NULL    |  
+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)  
  
mysql>  
  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type      | Null | Key | Default | Extra      |  
+-----+-----+-----+-----+-----+-----+  
| folio      | int(11)   | NO   | PRI | NULL    | auto_increment  
| id_usuario | char(20)  | YES  |     | NULL    |  
| modulo     | char(1)   | YES  |     | NULL    |  
| fecha_cont | date      | YES  |     | NULL    |  
| id_admin   | char(20)  | NO   |     | NULL    |  
+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```



```
mysql> describe rep_falla;
```

Field	Type	Null	Key	Default	Extra
salida_prob	date	NO		NULL	
id_admin	char(20)	YES		NULL	
num_serie	char(20)	YES		NULL	
descripcion_falla	char(50)	YES		NULL	
modulo	char(1)	YES		NULL	
fecha_rep	date	YES		NULL	
hora_rep	time	YES		NULL	
num_reporte	int(11)	NO	PRI	NULL	auto_increment

```
8 rows in set (0.00 sec)
```

```
mysql> show tables;
```

Tables_in_admonbicicletas
bici_administradores
bici_presdev
bici_usuarios
bicicletas
contrato_usu
facturas
proveedores
rep_falla

```
8 rows in set (0.00 sec)
```