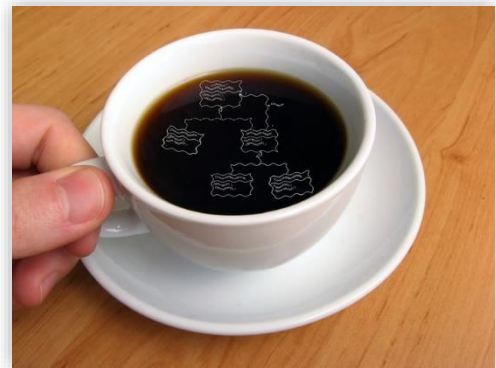


# LSTI1813 Modelación de Aplicaciones

## Estimado colega:

Como formadores de ingenieros en desarrollo de software, es primordial sentar las bases para que nuestros alumnos puedan alcanzar la competencia de desarrollar un modelo de aplicación para optimizar recursos mediante herramientas orientadas a objetos.

La modelación de aplicaciones es de gran utilidad porque, sin importar el lenguaje de desarrollo que se utilice, se podrán desarrollar aplicaciones que cumplan con los requisitos funcionales y no funcionales. Durante esta materia tendremos la oportunidad de enseñar las bases y estructuras para modelar en UML. A través de softwares como Lucidchart y Code2Flow, los alumnos podrán aprender haciendo. Code2Flow permitirá visualizar paso a paso las entradas, procesos y salidas de los modelos. Netbeans IDE 8.2 y TomCat 9.0 verificarán la ingeniería inversa de los modelos.



## Módulo 1. Conceptos fundamentales

Durante este módulo se profundizará sobre el modelado de aplicaciones, que permite representar de manera conceptual el comportamiento y las funcionalidades que intervienen en un negocio, para poder llegar a una mejor comprensión del dominio del problema y a su respectiva solución, enfocando el desarrollo de las aplicaciones a lo establecido en el modelo.

En este módulo se va a presentar una visión general para poder tener una idea del proceso a alto nivel, y más adelante se verán los pasos que componen cada fase. Los alumnos aprenderán las ventajas de utilizar el lenguaje de modelado gráfico UML usando una herramienta que les permitirá comunicar los conceptos y decisiones, entender el problema y una solución propuesta, así como generar los diagramas y documentación que describen el problema y la solución.

### Tema 1. Historia del modelado de aplicaciones

Este primer tema ayudará a plasmar las bases del modelado de aplicaciones; se debe dar un recorrido por la evolución del modelado, así como identificar los campos de aplicaciones del modelado y finalizar con la importancia de los modelos.

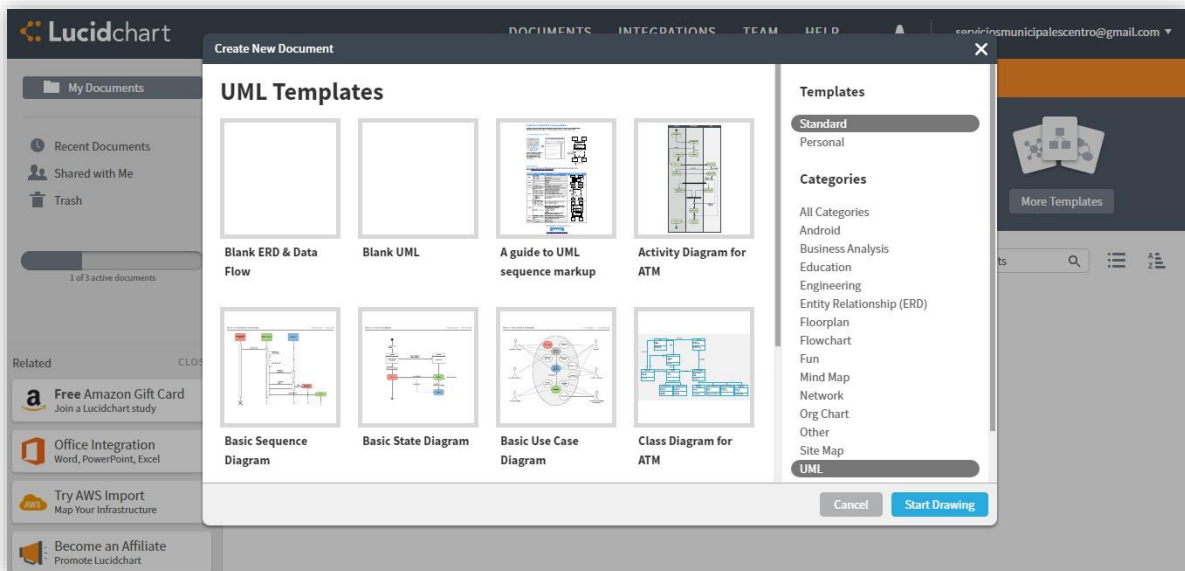
## Tema 2. Desarrollo de las aplicaciones

Este tema inicia con la definición del ciclo de vida de un proyecto y los diferentes tipos de ciclo de vida. Asegúrate de que el alumno identifique las características y elementos particulares de cada uno de ellos; en este tenor, se definirá qué es un modelo y las fases que se involucran en este proceso, para poder entrar de lleno a UML y a los diferentes tipos de diagramas. Para finalizar, se cerrará el tema con el desarrollo basado en componentes como recomendaciones para el modelado de aplicaciones.

### Actividad 1. Comprensión de los ciclos de vida para el desarrollo de aplicaciones

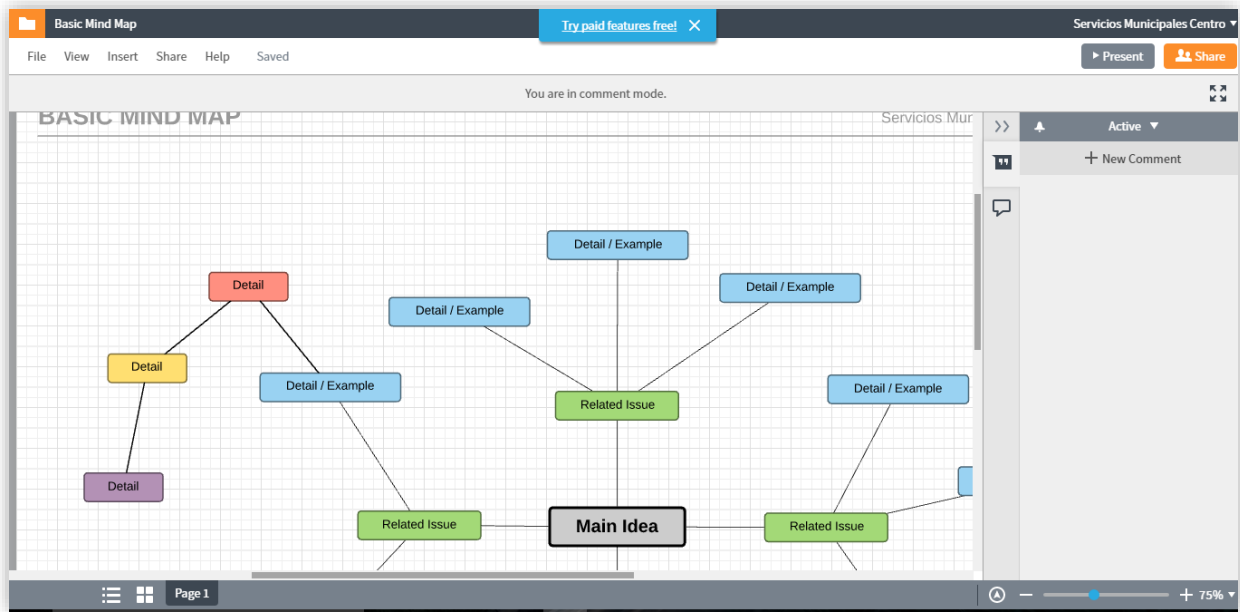
Con esta actividad los alumnos podrán comenzar a utilizar Lucidchart, identificando las diferentes categorías de diagramas.

**Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.**



Aprovechando la funcionalidad de mapa mental de Lucidchart, los alumnos podrán elaborar uno con los principios, características, personajes que intervinieron en la evolución y áreas de aplicación del modelado.

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.



Para la siguiente clase tendrás oportunidad de que tus alumnos pongan en práctica sus conocimientos, a través de la identificación de los diferentes tipos de ciclo de vida de un proyecto, seleccionando el más completo y describiendo sus principales ventajas y desventajas.

Para terminar, los alumnos deberán crear un blog cubriendo los siguientes elementos:

- Las fases para alcanzar los requerimientos de los clientes.
- Importancia de generar modelos.
- ¿Cuáles son los campos de aplicación del modelado?
- ¿Qué es el aspecto semántico?
- ¿A qué se refiere la notación?
- ¿Por qué las vistas son de utilidad?
- ¿Qué significa el desarrollo basado en componentes?
- Ventajas de revisar los diagramas del proyecto versus revisar el código de un sistema.
- Las bondades de utilizar UML.
- Consejos que se deben saber antes de empezar a modelar.

### **Tema 3. Concepto del paradigma orientado a objetos**

Durante este tema tus alumnos deberán comprender que un paradigma es una forma de entender y representar la realidad: un conjunto de teorías, estándares y métodos que, juntos, representan un modo de organizar el pensamiento, es decir, un modo de ver el mundo. Cada nuevo paradigma responde a una necesidad real de nuevas formas de afrontar problemas. A menudo, se crea un nuevo paradigma como respuesta a las deficiencias de paradigmas anteriores.

Un paradigma de programación es una forma de conceptualizar en qué consiste la ejecución de un programa y cómo deben estructurarse y organizarse las tareas que se llevarán a cabo en esa ejecución. El paradigma orientado a objetos (OO) define los programas en términos de comunidades de objetos. Los objetos con características comunes se agrupan en clases. Los objetos son entidades que combinan un estado y un comportamiento. Estos objetos se comunican entre ellos para realizar tareas. Las propiedades de la orientación a objetos son ciertas premisas o características especiales que son vistas como las ventajas o bondades del paradigma de orientación a objetos.

De igual manera, se introducirá la nomenclatura y los estereotipos del UML.

### **Tema 4. Lenguaje unificado de modelado**

Este tema permitirá a los alumnos comprender los conceptos y tipos de elementos, las relaciones y diagramas de UML, y aplicar las reglas y mecanismos comunes de UML.

De manera particular, debe quedar claro cuáles son los tipos de elementos (estructurales, de comportamiento, agrupación y anotación), ya que los alumnos empezarán a crear sus primeros diagramas en Code2Flow, y parte del éxito será que identifiquen los conceptos y características, más que utilizar el software.

### **Tema 5. Herramientas CASE para el modelado de aplicaciones**

Hoy en día existen herramientas que nos ayudan a automatizar tareas, facilitar la construcción de modelados y código con la finalidad de hacer más eficiente el desarrollo de aplicaciones. La tecnología denominada CASE (Computer Aided Software Engineering) se define como “el conjunto de herramientas y metodologías que soportan un enfoque de ingeniería para las distintas fases del desarrollo de software”. Durante este curso utilizaremos Code2Flow y LucidChart.

Es importante mostrar los campos de aplicación de las herramientas CASE y su clasificación. Para cerrar, dedica un tiempo de tu clase para mostrar los diferentes sitios web y herramientas útiles para el modelado, ya que servirán de apoyo para la realización de sus actividades y evidencias.

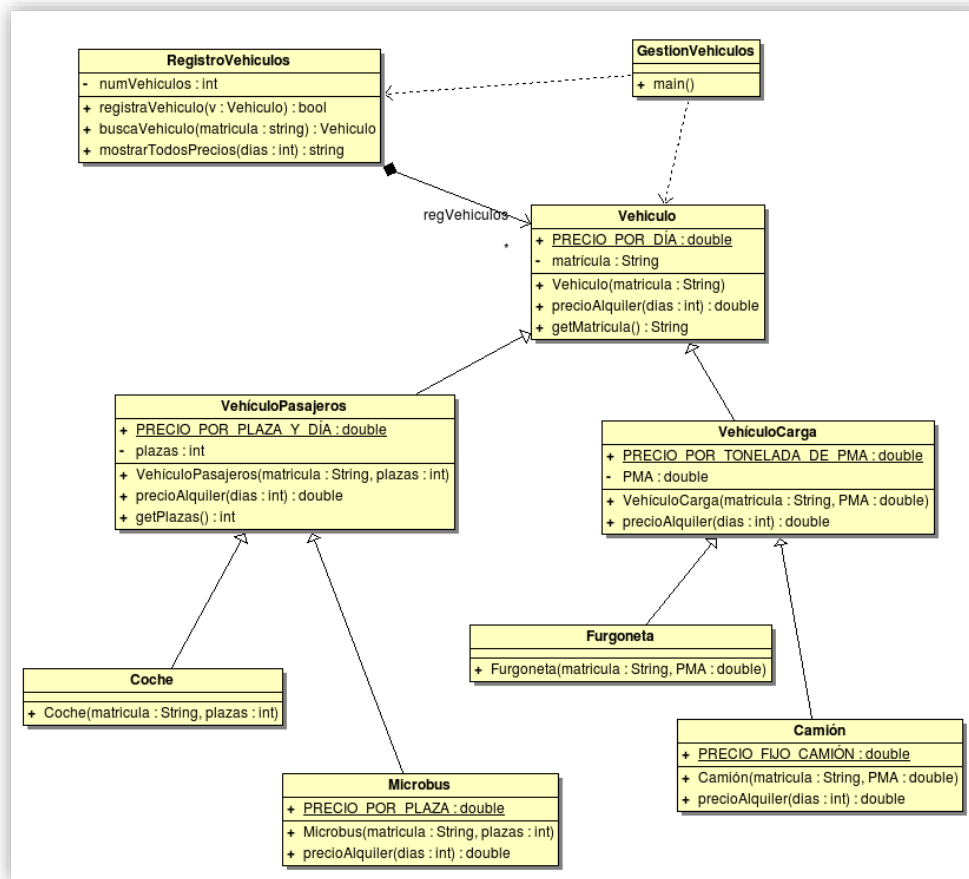
## Actividad 2. Modelado de situación real utilizando UML

Durante esta actividad, los alumnos podrán empezar a identificar la nomenclatura, relaciones y elementos del modelado.

Elemento	Ejemplo
Clase	Persona
Objeto	Juan
Atributos	Edad, estatura, peso
Valores de atributos	18 años, 1.60 m, 60 kg
Estado	Adulto, casado
Operación	Cumpleaños, teléfono
Parte del diseño	Base de datos, tabla

Después podrán aplicar sus conocimientos con un caso práctico, como el arrendamiento de vehículos, y generarán su primer diagrama de clases.

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.



## Módulo 2. Análisis y diseño de las aplicaciones

En este módulo guiarás a los alumnos en el proceso unificado de desarrollo de software, las disciplinas de requerimientos, análisis y diseño para proporcionar el detalle conceptual y técnico como punto de partida para la disciplina de implementación.

Es importante que proporciones la formación y experiencia práctica necesaria para analizar un problema y plantear una solución, centrándote en el uso adecuado de las tecnologías orientadas a objetos y el manejo de los pasos simplificados en el proceso de desarrollo de aplicaciones.

### Tema 6. Análisis orientado a objetos

Este módulo inicia introduciendo varios conceptos, entre los cuales están los involucrados, los FURPS, la trazabilidad y el modelado de casos de usos. Asimismo, los artefactos son visión, glosario, plan de administración, especificación de requerimientos, *storyboard* y especificación suplementaria.

Este tema también servirá para aprender sobre un nuevo diagrama, el modelado de casos de usos, por lo que es importante cuidar los elementos, las características, los flujos y relaciones.

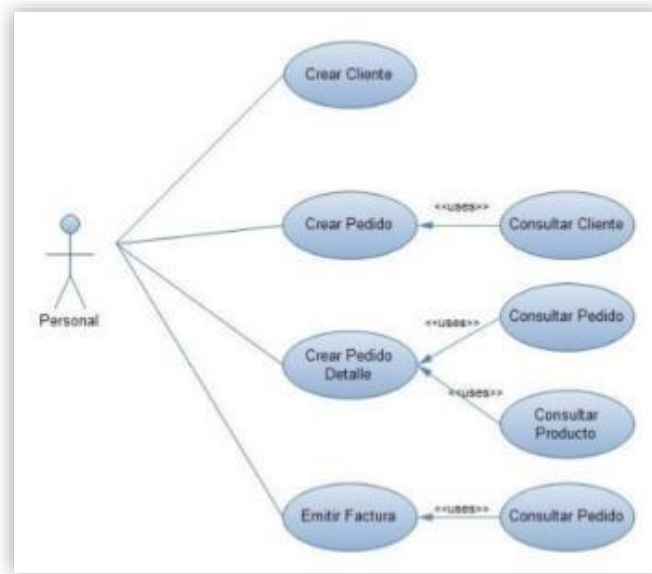
### Tema 7. Modelo de análisis

Aquí continuarás revisando las técnicas y componentes UML del modelado de casos de uso del tema anterior, profundizando en la nomenclatura y sugerencias para obtener casos de usos completos y la importancia de una adecuada documentación del mismo.

### Actividad 3. Aprendiendo a identificar requerimientos y construir casos de uso para la solución de un problema

Durante esta actividad los alumnos tendrán la oportunidad de poner en práctica sus aprendizajes sobre la ingeniería de requerimientos, diseñar el modelado de casos de uso y sus flujos.

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.



---

## Tema 8. Diseño orientado a objetos

Este tema es fundamental para la materia, ya que se desarrollarán el resto de los diagramas del modelado de aplicaciones; en general, la nomenclatura y las reglas revisadas en los temas anteriores ayudarán mucho para permitir centrarse en las particularidades de cada diagrama.

## Tema 9. Base de datos

Las bases de datos tienen una gran relevancia y son una de las mayores aportaciones que ha dado la informática a las empresas. Debes transmitirles a los alumnos que la gestión adecuada de una base de datos permite diferentes ventajas, por ejemplo, aumenta la eficacia (habrá trabajos que se realicen con mayor rapidez y agilidad debido a la simplificación de los mismos), cuida la seguridad de los datos que se almacenan y, con todos estos factores, se maximizan los tiempos y, por lo tanto, se produce una mejora en la productividad.

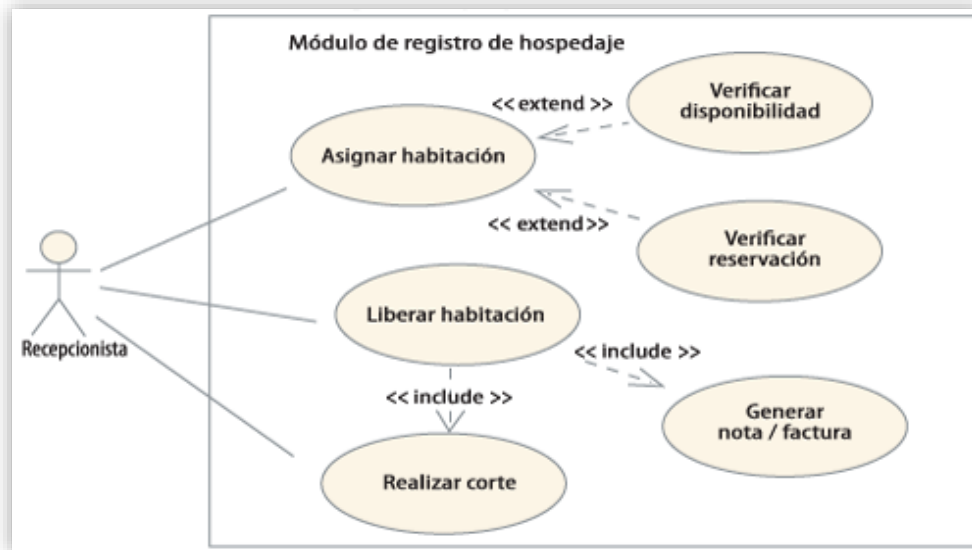
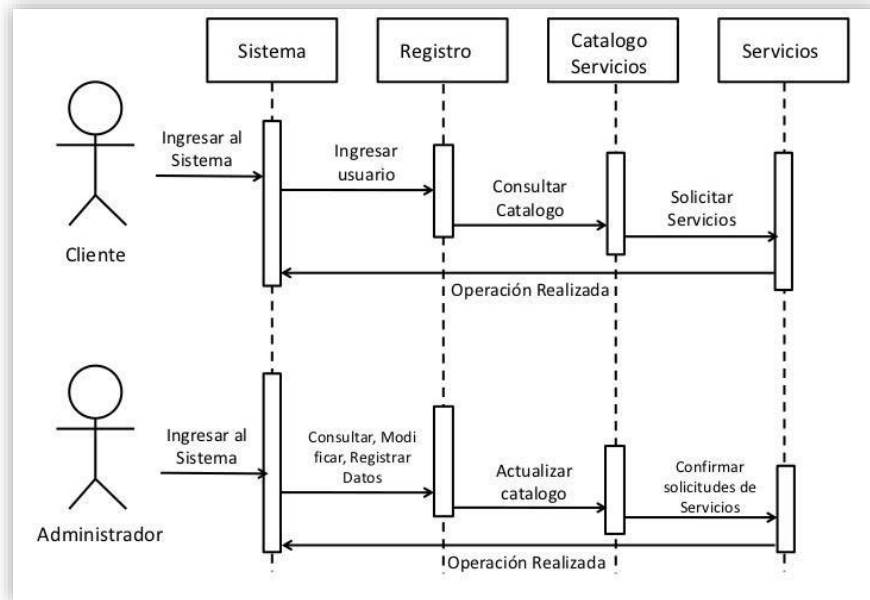
## Tema 10. Ingeniería de diseño

Este tema permitirá vislumbrar la importancia de cómo esbozar un diseño con un diagrama de clases y con un diagrama de interacción puede ayudar a poner en orden los pensamientos y a facilitar la codificación, así como todos los borradores y prototipos rápidos que servirán para hacer más fácil la generación del código.

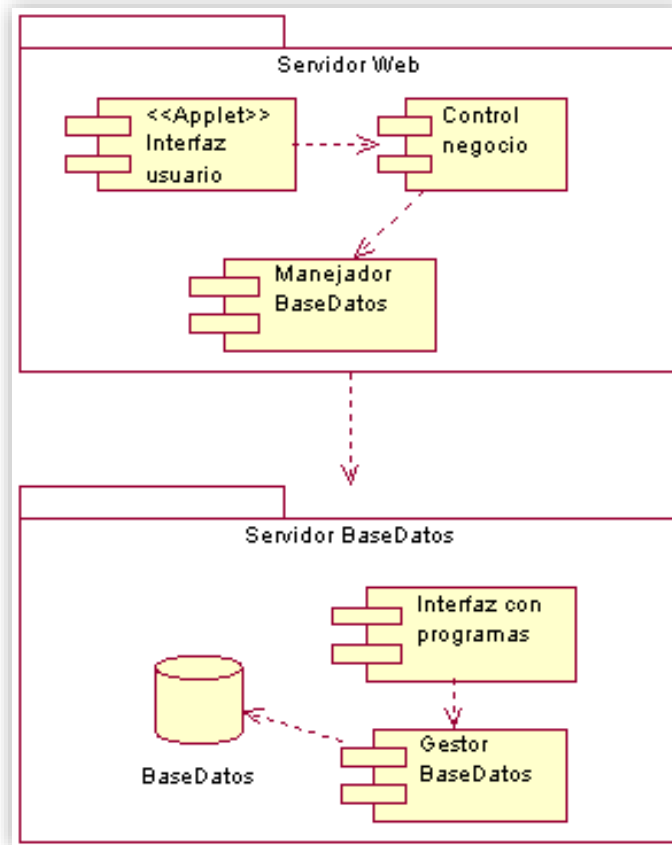
#### Actividad 4. Aprendiendo a construir diagramas y bases de datos

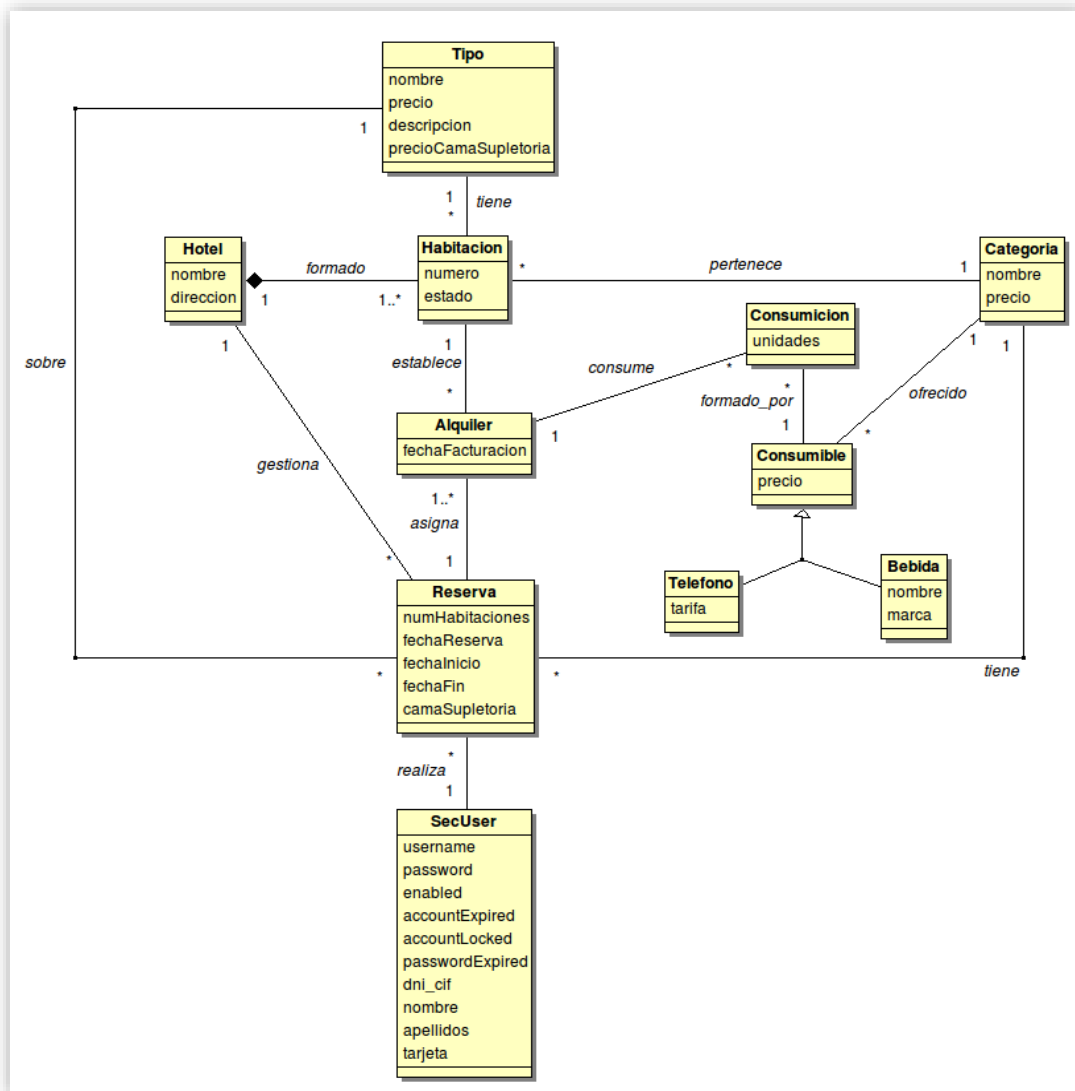
En esta actividad el alumno podrá realizar un modelado completo de una aplicación, diseñando los diferentes tipos de diagramas y el diseño de la base de datos.

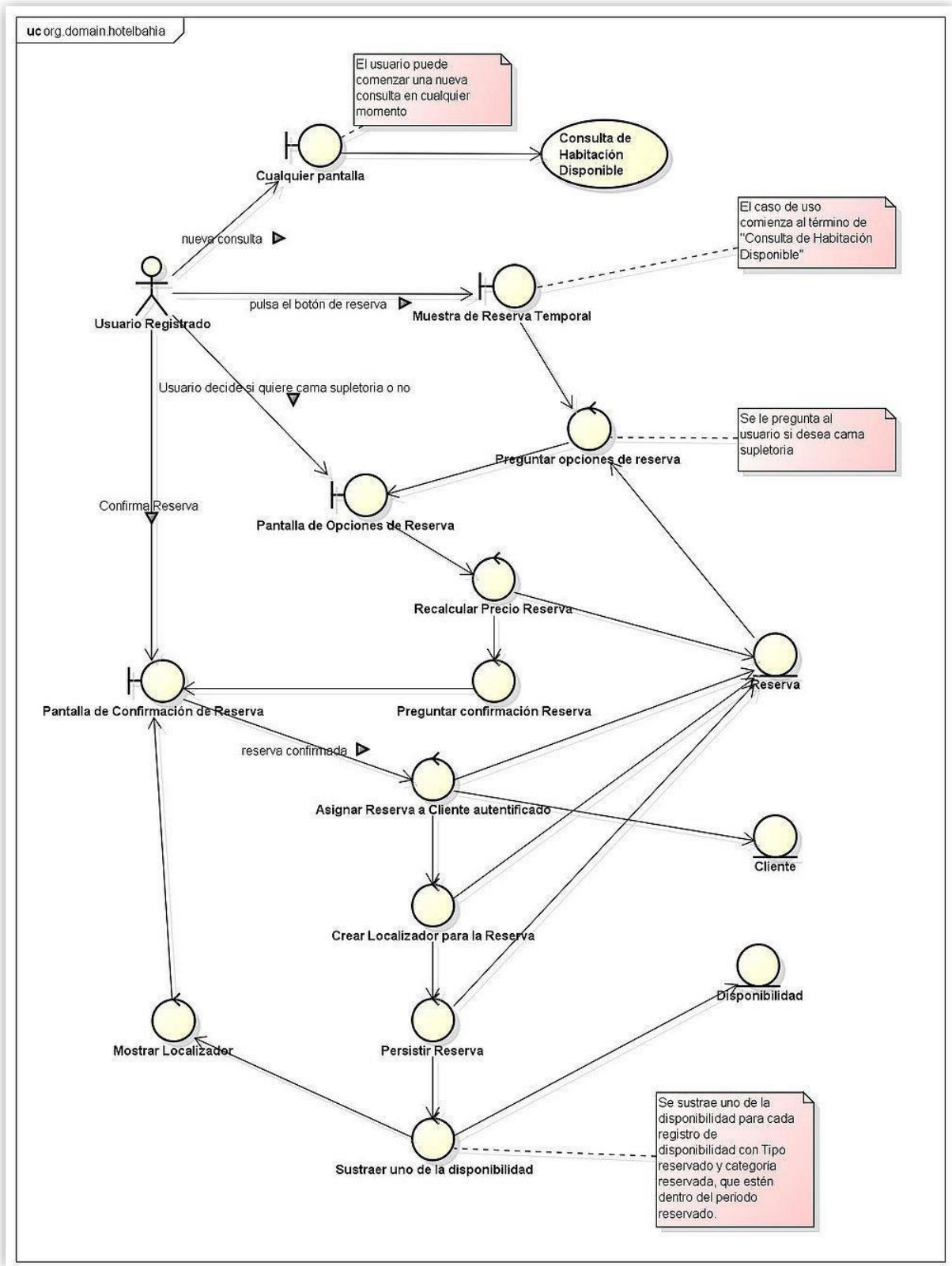
Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.



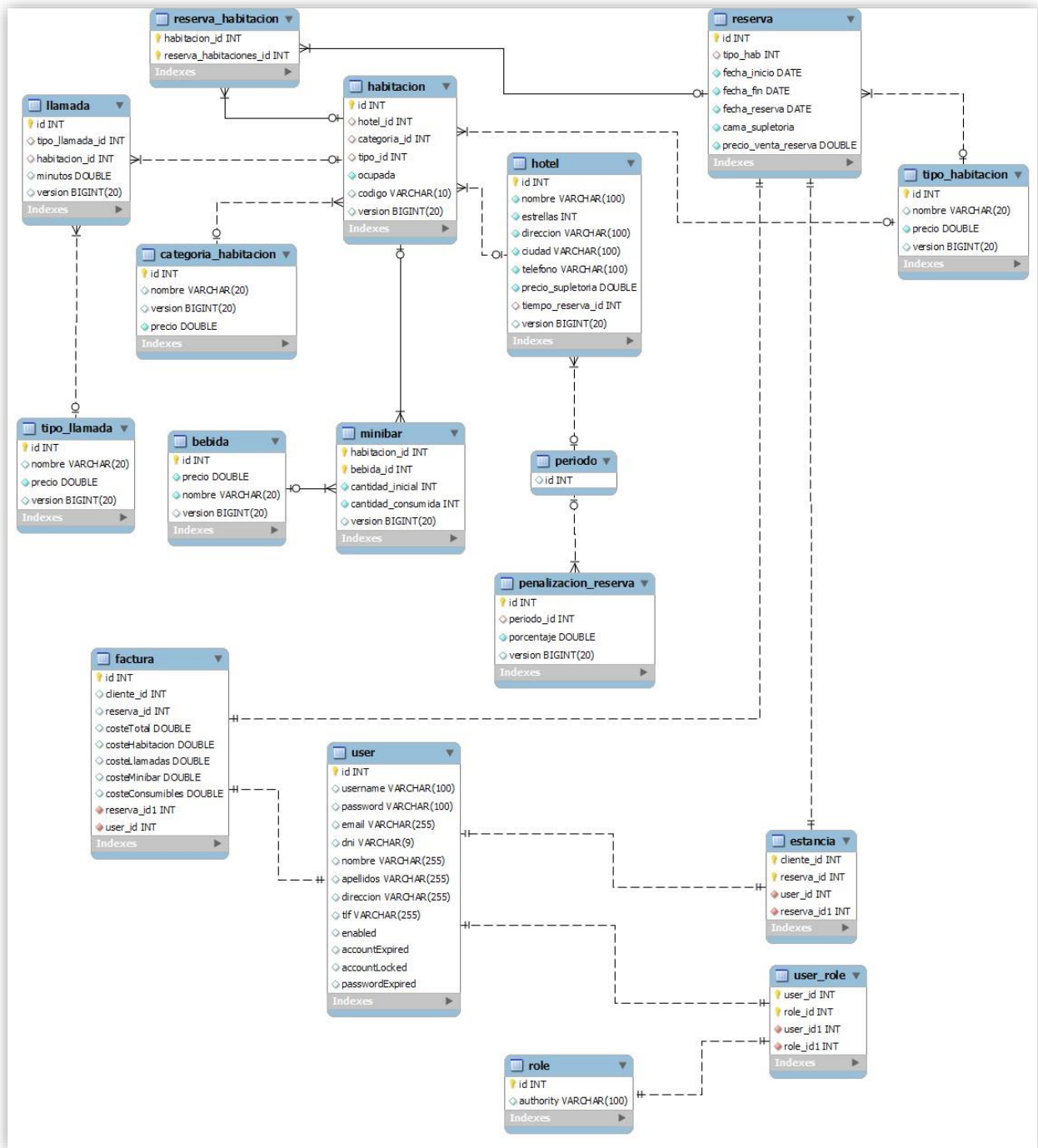








Diseño general de base de datos:



## Módulo 3. Construcción y despliegue de aplicaciones

Recuerda que la existencia de defectos en las aplicaciones impacta de manera significativa en su uso y en las operaciones de una empresa o institución, provocando problemas con el servicio, inconsistencias en la información que se genera o manipula, y pérdidas monetarias. Por ello, es de suma importancia planear, diseñar y ejecutar pruebas en las aplicaciones que permitan identificar defectos en las mismas y asegurar que funcionan de manera adecuada; mientras que las actividades de verificación y validación ayudan a revisar que las aplicaciones y otros entregables cumplan con los requisitos establecidos y la calidad necesaria para su correcto funcionamiento.

### Tema 11. La fase de la implementación

En este tema se abordarán los conceptos clave y las actividades de la fase de implementación, así como su relación con las otras fases.

### Tema 12. Plataformas de desarrollo

Aquí se abordarán las diferentes plataformas de desarrollo, los IDE y los lenguajes de programación más populares.

Durante este tema deberás plasmar las bases de la programación en JAVA, para que los alumnos puedan interpretar el código de la ingeniería inversa obtenido a partir de sus diagramas.

### Actividad 5. Comprensión de la fase de implementación y análisis de plataformas de desarrollo

Con esta actividad los alumnos podrán comprender la fase de implementación y analizar las diferentes características de las plataformas de desarrollo, los IDE y los lenguajes de programación.

En este tenor, los alumnos podrán programar su primera aplicación, pero deberás guiarlos en el proceso.

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.



---

### Tema 13. La fase de pruebas

En un proyecto de desarrollo de software se pueden presentar errores o bugs en cualquiera de las etapas del ciclo de vida del software; sin embargo, aun cuando se intente detectarlos después de cada fase utilizando técnicas como la inspección, algunos errores permanecen sin ser descubiertos, por ello, es necesario ofrecer la metodología y las técnicas que ayuden a los alumnos en este proceso, además de darles recomendaciones sobre cuándo utilizar cada tipo de pruebas.

### Tema 14. Verificación y validación

El objetivo que persiguen las actividades de verificación y validación es asegurar que los productos de trabajo cumplan con sus requerimientos especificados y demostrar que una aplicación o producto se ajusta a su uso contemplado cuando se sitúa en su entorno previsto.

En este tema se explicarán los conceptos de verificación y validación, y las diferencias que existen entre ellos.

### Tema 15. Fase de despliegue

Para cerrar el curso debes abordar el último diagrama del modelado, el diagrama de despliegue, con sus diferentes elementos, características y similitudes con el resto de los

Diagramas, así como las diferencias entre el modelado de un sistema empujado y uno cliente servidor.

### Actividad 6. Comprensión de la fase de pruebas, despliegue y verificación y validación

Con esta actividad se busca que los alumnos sean capaces de identificar las características de la fase de pruebas, la metodología, los tipos de pruebas que existen y para qué tipos de proyectos se debe realizar cada una. Además, los alumnos deben identificar las principales tareas de la fase de pruebas y de qué manera ayuda utilizar herramientas para realizar las actividades de la fase de pruebas, las actividades del proceso de verificación y validación, y el diagrama de despliegue.

**Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.**

