**Universidad de oriente**

**Núcleo de Anzoátegui**

**Escuela de Ingeniería y Ciencias Aplicadas**

**Departamento de Computación y Sistemas**

**Asignatura: Desarrollo de software**

**Profesor: Alumno:**

**Luis Correa CI: 19.840.230**

**Víctor Mujica**

**Sección: 01**

**Barcelona, 11 de Junio del 2013**

**Vistas de UML:**

No hay una línea clara entre los diversos conceptos y construcciones en UML, pero, por comodidad, nosotros los dividimos en varias vistas. Una vista simplemente es un subconjunto de las construcciones de modelado de UML que representan un aspecto del sistema. La división en diferentes vistas es un tanto arbitraria, pero esperamos que sea intuitiva. Uno o dos tipos de diagramas proporcionan una notación visual para los conceptos de cada vista.

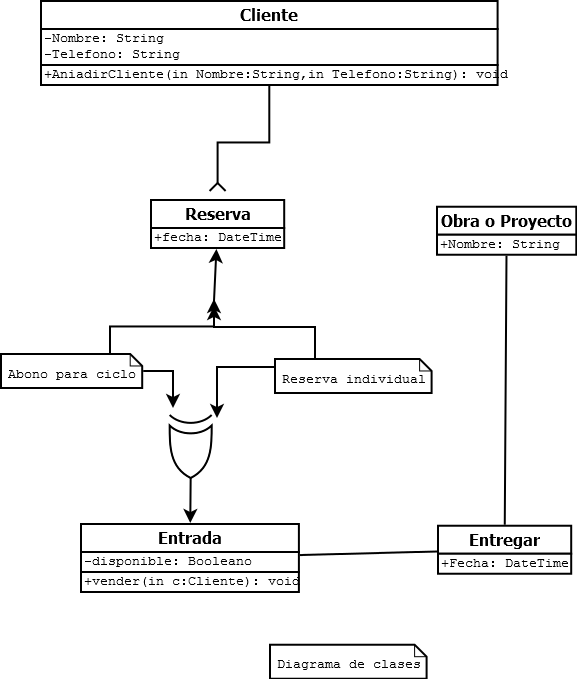
En el nivel superior, las vistas pueden ser divididas en las siguientes áreas: clasificación estructural, comportamiento dinámico, diseño físico y gestión del modelo.

**Vista estática:**

La vista estática modela conceptos del dominio de la aplicación, así como los conceptos internos inventados como parte de la implementación de una aplicación. Esta vista es estática porque no describe el comportamiento dependiente del tiempo del sistema, que se describe en otras vistas. Los principales componentes de la vista estática son las clases y sus relaciones: asociación, generalización y varios tipos de dependencia, como la realización y el uso. Una clase es la descripción de un concepto del dominio de la aplicación o del dominio de la solución. Las clases son el centro, alrededor del cual se organiza la vista de clases; otros elementos pertenecen o se unen a las clases. La vista estática se muestra en los diagramas de clases, llamadas así porque centran fundamentalmente su atención en las clases.

Las clases se dibujan como rectángulos. La lista de atributos y operaciones se muestran en compartimentos separados. Se pueden suprimir los compartimentos cuando no se necesitan todos los detalles. Una clase puede aparecer en varios diagramas. Los atributos y operaciones se muestran a menudo en un diagrama (el diagrama “raíz”) y se suprimen del resto de los diagramas.

Las relaciones entre clases se muestran como las líneas que conectan los rectángulos de las clases. Los diferentes tipos de relaciones se distinguen por la textura de la línea y por los adornos en las líneas o en sus extremos.

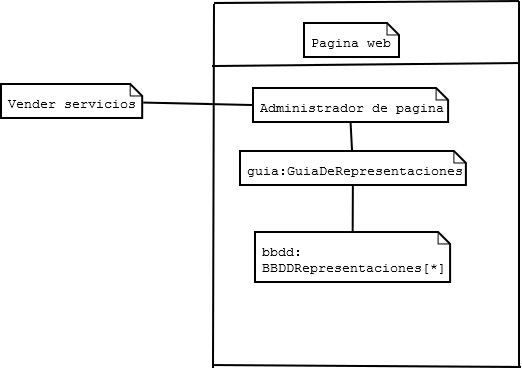


**Vistas de diseño:**

Las vistas previas modelan los conceptos de la aplicación desde un punto de vista lógico. Las vistas de diseño modelan la estructura de diseño de la propia aplicación, como su expansión en clasificadores estructurados, las colaboraciones que proporcionan funcionalidad y su ensamblado a partir de componentes con interfaces bien definidas. Estas vistas proporcionan una oportunidad para establecer una correspondencia entre las clases y los componentes de implementación, y expandir las clases de alto nivel en una estructura de soporte. Los diagramas de implementación incluyen el diagrama de estructura interna, el diagrama de colaboración y el diagrama de componentes.

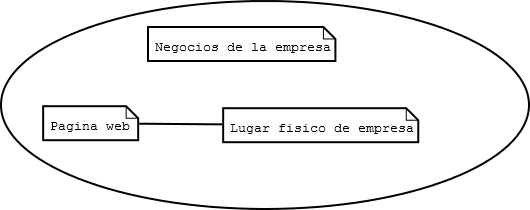
**Diagrama de estructura interna:**

Una vez que comienza el proceso de diseño, las clases se deben descomponer en colecciones de partes conectadas que, posteriormente, se deben descomponer por turnos. Un clasificador estructurado modela las partes de una clase y sus conectores contextuales. Una clase estructurada puede ser encapsulada forzando a que las comunicaciones desde el exterior pasen a través de los puertos cumpliendo con las interfaces declaradas. Un diagrama de estructura interna muestra la descomposición de una clase.



**Diagrama de colaboración:**

Una colaboración es una relación contextual entre un conjunto de objetos que trabajan juntos para lograr un propósito. Contiene una colección de roles —ranuras contextuales dentro de un patrón genérico, que pueden ser representadas por objetos individuales, o vinculadas a ellos.



**Diagrama de componentes:**

Un componente es un tipo de clasificador estructurado, por lo que debe definirse su estructura interna en un diagrama de estructura interna.

