1)1. **Sistema de tipos:** es un conjunto de reglas usadas por un lenguaje para estructurar y organizar sus tipos

Principal función: escribir programas seguros.

2. **Tipado fuerte:** el sistema de tipos es fuerte cuando especifica restricciones sobre como las operaciones que involucran valores de diferentes tipos pueden operarse.

Tipado débil: lo contrario al anterior.

Tipado debil

```
a = 2
b= "2"
Concatenar (a,b) //retorna "22"
Sumar (a,b) //retorna 4
```

tipado fuerte

```
a = 2
b= "2"
Concatenar (a,b) //error de tipos
Sumar (a,b) //error de tipos
Concatenar (str(a),b) //retorna "22"
    Sumar (a,int(b)) //retorna 4
```

3.**Tipado estático:** se realiza la ligadura en compilación, exige:

- Utilizar tipos de datos predefinidos
- Todas las variables se declaran con un tipo asociado
- Todas las operaciones se especifican indicando los tipos de los operandos requeridos y el tipo del resultado

Tipado dinámico: se realiza la ligadura en ejecución (provoca más comprobaciones en tiempo de ejecución (falta ejemplos)

2)**Tipos de datos**

1.Un tipo de dato es un conjunto de valores y un conjunto de operaciones que se pueden utilizar para manipularlos.

2.Un tipo predefinidio elemental es un tipo de dato que es indivisible, no puede descomponerse a partir de otros. Ejemplo: enteros, reales, caracteres, booleanos.

3.Un tipo definidio por el usuario es un dato definido en función de una agrupación de objetos de datos elementales. Ejemplo: enumerados, arreglos, registros, listas, etc.

3)**Tipos compuestos:**

1. - Producto Cartesiano: el producto cartesiano de n conjuntos A1, A2, .. An denotado A1XA2X...An es un conjunto donde sus elementos estan ordenados en n-tuplas (a1, a2,..an) donde cada ai pertenece a Ai. Por ejemplo un registro.

-Correspondencia finita: función de un conjunto finito de valores de un tipo de dominio DT en valores de un tipo del dominio RT.

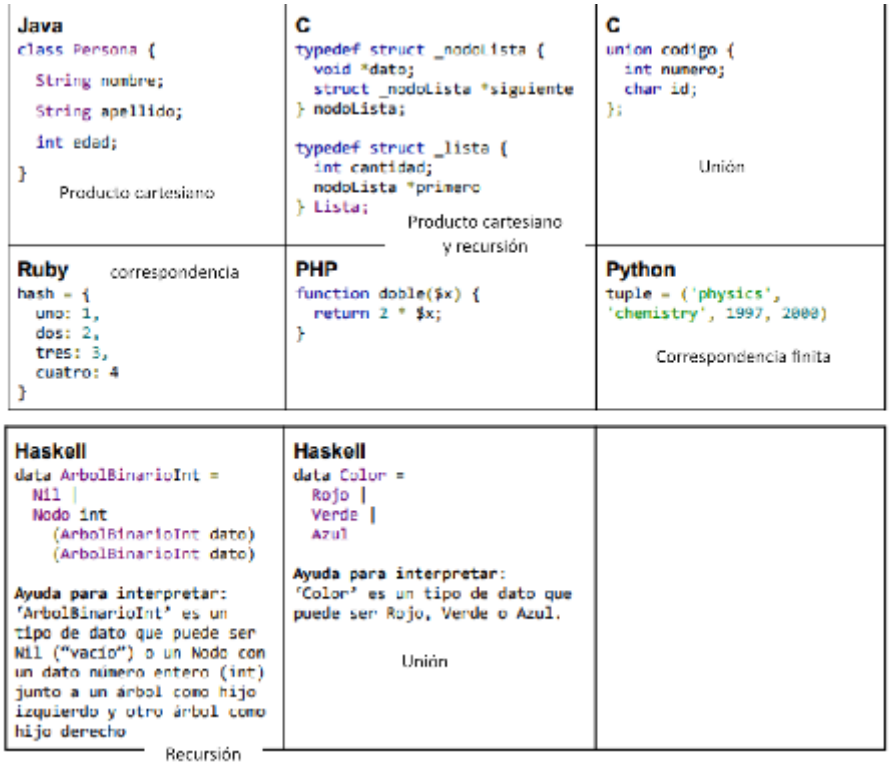
Define un mapeo entre los valores de DT de li a ls hacia valores de RT. Por ejemplo arreglos, vectores y matrices.. Estan indexados,, ordenados.

-Unión y unión discriminad.a:

- de 2 o mas tipos define un tipo como la disjunción de los tipos dados
- Permite manipular diferentes tipos en distintos momentos de la ejecución
- Chequeo dinámico
- Declaración : union address{//campos mutuamente short int offset; long unsigned int absoluto; };

-Tipos recursivos:

- Un tipo de dato recursivo T se define como una estructura que puede contener componentes del tipo T
- Define datos agrupados donde su tamaño puede crecer arbitrariamente y su estructura puede ser arbitrariamente compleja.
- Los lenguajes convencionales soportan esta implementacion a través de punteros(ref. A un objeto).
- Ejemplos:árboles, listas de Pascal



2.

4.**Mutabilidad/Inmutabilidad**

1.Mutabilidad: aquel dato que puede cambiar luego de ser creado.

Inmutabilidad: aquel dato que no puede cambiar luego de ser creado.

Ejemplos inmutables:

Python : enteros, cadenas y tuplas

Ruby: casi todos son mutables pero con el método freeze se vuelven inmutables(no se puede modificar)

Ejemplos mutables:

Python: listas, conjuntos y diccionarios.

2. la capacidad de reasignar a a un nuevo objeto no afecta la mutabilidad del objeto original creado con Dato.new(1). Para determinar la mutabilidad de ese objeto, necesitaríamos conocer más detalles sobre la implementación de la clase Dato.

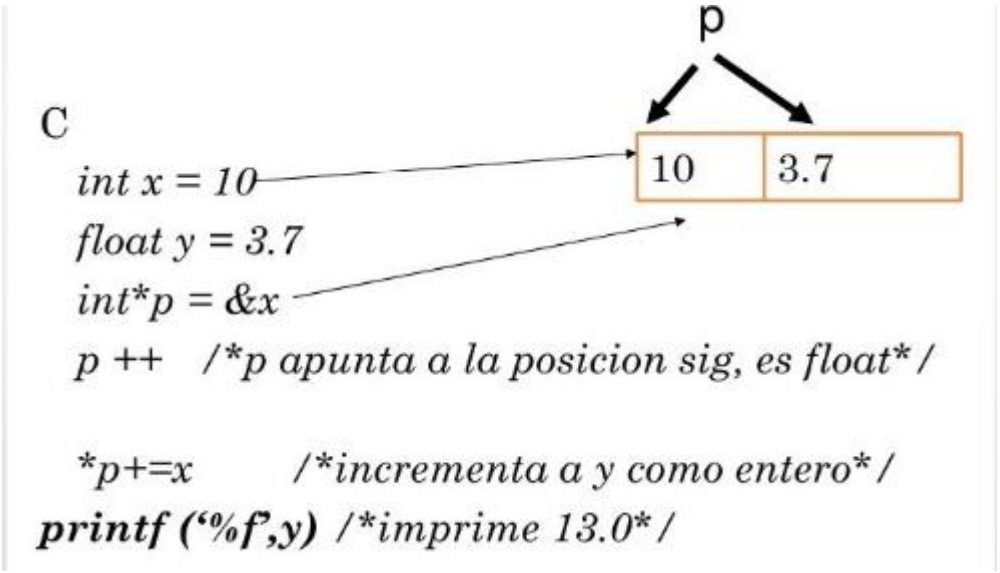
En resumen, basándonos únicamente en el código proporcionado, no podemos determinar si el objeto Dato.new(1) es mutable o no. La mutabilidad de dicho objeto

dependerá de la implementación interna de la clase Dato y de si los valores almacenados en ese objeto pueden cambiar después de su creación.

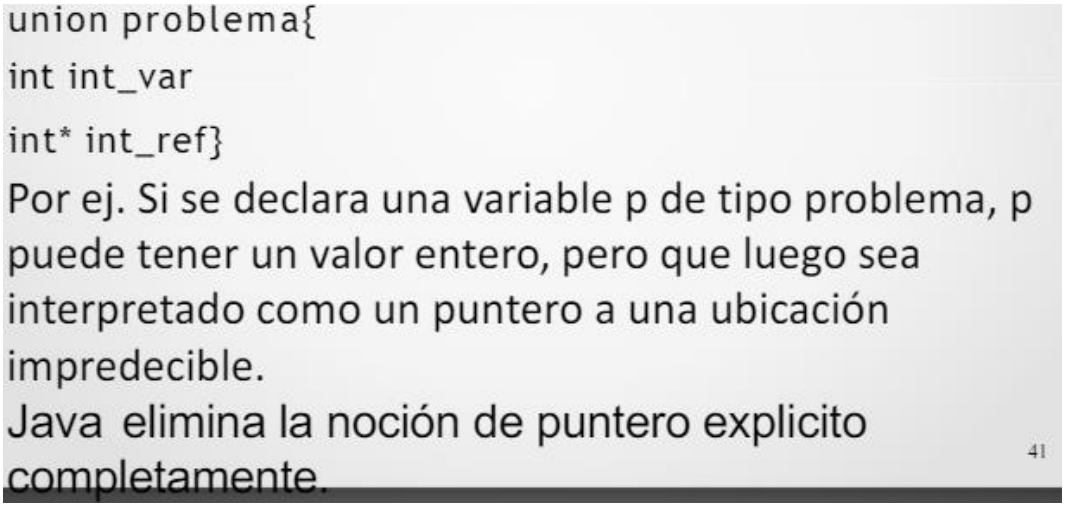
5.**Manejo de punteros**

1. C permite tomar el l-valor(dirección de memoria) de las variables usando &

2.Problemas en el manejo de punteros:

- Violación de tipos
  - 

- Referencias sueltas- referencias dangling
  - Si el objeto no esta alocado el puntero es dangling(peligroso)
  - Referencia suelta=> puntero con una dirección de una variable dinámica desalocada. Si se intenta usar el puntero genera error
- Punteros no inicializados
  - Peligro de acceso descontrolado a posiciones de memoria
  - Verificación dinámica de la inicialización
  - Se soluciona con el valor especial nulp
- Punteros y uniones discriminadas



- Alias
  - Cuando 2 o + punteros apuntan a la misma posición
- Liberación de memoria: objetos perdidos
  - Las variables de tipo puntero se alocan como cualquier otra en la pila del R.A

6.**TAD**

1.Una unidad para ser un TAD debe cumplir..

- Encapsulamiento:
  - La representación del tipo y las operaciones permitidas para los objetos de ese tipo, se describen con una unica unidad sintáctica.
  - Refleja las abstracciones descubiertas en el diseño
- Ocultamiento de la información:
  - La representación de los objetos y la implementación del tipo permanecen ocultos.
  - Refleja los niveles de abstracción

2.Ejemplos:

- **Java:** ArrayList, LinkedList, HashMap, TreeSet, PriorityQueue
- **Python:** list, tuple, set, dictionary, deque
- **C++:** std::vector, std::map, std::queue, std:: stack**.**
- **ADA:** Ada.Containers.Vectors, Ada.Containers.Hash_tables, Ada.containers.Linked_Lists, Ada.Containers.Priority_Queue.