



Práctica 5: Datatypes (continuación)

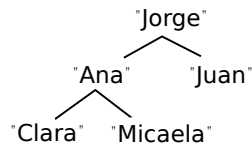
En esta práctica vamos a ver distintos datatypes que sirven para representar árboles. ¿Qué tipo árbol conviene usar? En realidad no hay una respuesta única, dependerá del problema particular que se nos presente.

1. Considere la siguiente definición de árbol:

```
- datatype 'etiqueta arbolbin =  
    Vacio |  
    Nodo of 'etiqueta * 'etiqueta arbolbin * 'etiqueta arbolbin;  
datatype 'a arbolbin = Nodo of 'a * 'a arbolbin * 'a arbolbin | Vacio
```

y responda a las siguientes consignas:

- Dé 1 ejemplo de un elemento que pertenezca a este tipo de dato.
- Represente el siguiente árbol usando este datatype.



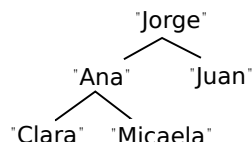
- Escriba una función que tome un árbol de este tipo y un elemento x y diga si x aparece o no en el árbol.
- Escriba una función que tome un árbol de este tipo y un elemento x y devuelva la cantidad de veces que x aparece en el árbol.
- Escriba una función que tome como argumento un árbol de tipo `int arbolbin` y devuelva la suma de los valores de sus nodos.
- Escriba una función que tome como argumento un árbol de tipo `int arbolbin` y devuelva el producto de los valores de sus nodos.
- Escriba una función que tome como argumento un árbol de tipo `int arbolbin` y devuelva otro `int arbolbin` donde los valores de sus nodos sean el doble de los valores de los nodos del árbol de entrada.

2. Considere la siguiente definición de árbol:

```
- datatype ('etiqueta) arbol1 =  
    Nodo of 'etiqueta * 'etiqueta arbol1 list;  
datatype 'a arbol1 = Nodo of 'a * 'a arbol1 list
```

y responda a las siguientes consignas:

- Dé 1 ejemplo de un elemento que pertenezca a este tipo de dato.
- Represente el siguiente árbol usando este datatype.



- Escriba una función que tome un árbol de este tipo y un elemento x y diga si x aparece o no en el árbol.
- Escriba una función que tome un árbol de este tipo y un elemento x y devuelva la cantidad de veces que x aparece en el árbol.
- Escriba una función que tome como argumento un árbol de tipo `int arbolb1` y devuelva la suma de los valores de sus nodos.

- f. Escriba una función que tome como argumento un árbol de tipo `int arbol1` y devuelva el producto de los valores de sus nodos.
 - g. Escriba una función que tome como argumento un árbol de tipo `int arbol1` y devuelva otro `int arbol1` donde los valores de sus nodos sean el doble de los valores de los nodos del árbol de entrada.
- 3. Consideremos nuevamente el tipo de dato `'a arbolbin` y supongamos que para el tipo de dato `'a` existe una relación de orden $<$. Implemente las siguientes funciones:
 - a. Implemente una función `busqueda` que reciba un `'a arbolbin` que sea un árbol binario de búsqueda y un elemento de tipo `'a` y devuelva `true` si dicho elemento pertenece al árbol, `false` en caso contrario.
 - b. Implemente una función `insertar` que reciba un `'a arbolbin t` que sea un árbol binario de búsqueda y un elemento x de tipo `'a` y devuelva otro árbol de búsqueda que se obtenga de insertar el elemento x en t .