```
-- a) Reglas de tipado
                                Γ ⊢ M1 : t  Γ ⊢ M2 : [t]
___________t-vacia              ________________________t-lista
Γ ⊢ []t: [t]                    Γ ⊢ M1 :: M2 : [t]


Γ ⊢ M : [t]                  Γ ⊢ M : [t]
_____________t-head          _____________t-tail
Γ ⊢ head(M) : t              Γ ⊢ tail(M) : [t]


-- b) Valores:
V ::= ... | []t | V :: V

-- Reglas de Reducción:

      M -> M'                          M2 -> M2'
___________________e-lista1      _________________e-lista2
M :: M2 -> M' :: M2              V :: M2 -> V :: M2'



     M -> M'
_________________e-head1         _________________e-head2
head(M) -> head(M')              head(V:: []t) -> V

_____________________________________________________________e-head3
head(V1 :: V2 :: V3) -> if V1 < V2 then head(V1 :: V3)
                                   else head(V2 :: V3)


     M -> M'
_________________e-tail1         _________________e-tail2
tail(M) -> tail(M')              tail(V::[]t) -> []t


_____________________________________________________________e-tail3
tail(V1 :: V2 :: V3) -> if V1 < V2 then V2 :: tail(V1 :: V3)
                                   else V1 :: tail(V2 :: V3)


c) head ((λx:Nat. 2) Zero :: 1 :: []nat)  --> e-head1, β, e-lista1
   head(2 :: 1 :: []nat)  --> e-head3
   if 2 < 1 then head(2 :: []nat) else head(1 :: []nat) --> if-false
   head(1 :: []nat)    --> e-head2
   1.
```