

```

1. #define FALSE 0
   #define TRUE 1
   sem mutex; mutex.valor = 1;
   int cruzandoOeste = FALSE; int cruzandoLeste = FALSE;

   CruzaOeste()                CruzaLeste()
   {                            /* simetrico a CruzaOeste() */
       int consegui = FALSE;
       repeat
           P(mutex)
           if (!cruzandoLeste){
               cruzandoOeste = TRUE;
               consegui = TRUE;
           }
           V(mutex);
       until (consegui);
       Cruza();
       P(mutex);
       cruzandoOeste = FALSE;
       V(mutex);
   }

```

```

2. sem oeste; oeste.valor = 0;
   sem leste; leste.valor = 1;

   CruzaOeste()                CruzaLeste()
   {                            {
       P(oeste);                P(leste)
       Cruza();                 Cruza();
       V(leste);                V(oeste);
   }                            }

```

```

3. int cruzandoLeste = 0; int queremLeste = 0;
   int cruzandoOeste = 0; int queremOeste = 0;
   sem oeste; oeste.valor = 0;
   sem leste; leste.valor = 0;
   sem mutex; mutex.valor = 1;

   CruzaOeste() {                CruzaLeste() {
       P(mutex);                P(mutex);
       cruzandoOeste++;         cruzandoLeste++;
       if (!cruzandoLeste)     if (!cruzandoOeste)
           V(oeste);           V(leste);
       else                     else
           queremOeste++;     queremLeste++;
       V(mutex);                V(mutex);
       P(oeste);                P(leste);

       Cruza();                 Cruza();

       P(mutex);                P(mutex);
       cruzandoOeste--;         cruzandoLeste--;
       if (!cruzandoOeste && queremLeste)
       {                         if (!cruzandoLeste && queremOeste)
           {                       {
               queremLeste--;     queremOeste--;
               V(leste);         V(oeste);
           }                       }
       }                          }
       V(mutex);                V(mutex);
   }                              }

```