Apéndice B

Estrategias Alternativas

En cada caso, se tiene que

- F es un trozo de tamaño $O(K)$ de memoria local al procesador.
- H es el tamaño de la tabla de hashing utilizada para los locks.
- L_t lista invertida de un término.
- *tid* identificador del thread.
- $pair(d, f_t)$ es un ítem de lista invertida, donde f_t es la frecuencia del término t en el documento d.
- *lockRead(t)* permite a todos los lectores entrar en una zona protegida y *lockWrite(t)* espera a que todos los que están en la zona protegida terminen y luego el thread entra en modo exclusivo.
- *lock(t)* es un lock de acceso exclusivo.

```
global: q_write = false

while( true )

   lock( input queue )
      if q_write == false then
         Tn = extractNextTransaction( input queue )
      endif
      if Tn.type != QUERY then
         q_write = true
      endif
   unlock( input queue )

   if Tn.type == QUERY then /*read transaction*/

      for each term t in Tn.query do
         rank( L_t )
      endfor

   else /*write transaction*/

      Barrier()
         for each term t in Tn.termsForThread[tid] do
            d = Tn.docId
            if Tn.type == UPDATE then
               update( L_t, pair( d, f_t ) )
            else
               insert( L_t, pair( d, f_t ) )
            endif
         endfor
         if pid == 0 then q_write = false
      Barrier()
   endif
endwhile
```

Figura B.1: Pseudo-Código CR.

```
while ( true )

   lock( input queue )
      Tn = extractNextTransaction( input queue )

   if Tn.type == QUERY then /*read transaction*/

      for each term t in Tn.query do
         lockRead( t%H )
      endfor
      unlock( input queue )

      for each term t in Tn.query do
         rank( L_t )
         unlockRead( t%H )
      endfor

   else /*write transaction*/

      for each term t in Tn.terms do
         lockWrite( t%H )
      endfor
      unlock( input queue )

      for each term t in Tn.terms do
         d = Tn.docId
         if Tn.type == UPDATE then
            update( L_t, pair( d, f_t ) )
         else
            insert( L_t, pair( d, f_t ) )
         endif
         unlockWrite( t%H )
      endfor

   endif
endwhile
```

Figura B.2: Pseudo-Código TLP1.

```
while ( true )

   lock( input queue )
      Tn = extractNextTransaction( input queue )

   if Tn.type == QUERY then /*read transaction*/

      for each term t in Tn.query do
         lock( t%H )
      endfor
      unlock( input queue )

      for each term t in Tn.query do
         rank( L_t )
         unlock( t%H )
      endfor

   else /*write transaction*/

      for each term t in Tn.terms do
         lock( t%H )
      endfor
      unlock( input queue )

      for each term t in Tn.terms do
         d = Tn.docId
         if Tn.type == UPDATE then
            update( L_t, pair( d, f_t ) )
         else
            insert( L_t, pair( d, f_t ) )
         endif
         unlock( t%H )
      endfor

   endif
endwhile
```

Figura B.3: Pseudo-Código TLP2.

```
while( true )

   lock( input queue )
      Tn = extractNextTransaction( input queue )
   unlock( input queue )

   if Tn.type == QUERY then /*read transaction*/

      for each term t in Tn.query do
         for each block b in L_t do
            lock( t%H )
               F = fetch(L_t[b])
            unlock( t%H )
            rank( F )
         endfor
      endfor

   else /*write transaction*/

      for each term t in Tn.terms do
         d = Tn.docId

         lock ( t%H )
            if Tn.type == UPDATE then
               update( L_t, pair( d, f_t ) )
            else
               insert( L_t, pair( d, f_t ) )
            endif
         unlock ( t%H )

      endfor
   endif
endwhile
```

Figura B.4: Pseudo-Código RTLP.

```
while( true )

   lock( input queue )
      Tn = extractNextTransaction( input queue )
   unlock( input queue )

   if Tn.type == QUERY then /*read transaction*/

      for each term t in Tn.query do
         for each block b in L_t do
            lock( b%H ); lock( (b+1)%H )
               F = fetch(L_t[b])
            unlock( (b+1)%H ); unlock( b%H )
            rank( F )
         endfor
      endfor

   else /*write transaction*/

      for each term t in Tn.terms do
         d = Tn.docId

         for each block b in L_t do
            lock ( b%H ); lock ( (b+1)%H )

            if isRightBlock( b, f_t ) == true then
               if Tn.type == UPDATE then
                  update( b, pair( d, f_t ) )
               else
                  insert( b, pair( d, f_t ) )
               endif
               unlock ( (b+1)%H ); unlock ( b%H )
               next term
            endif

            unlock ( (b+1)%H ); unlock ( b%H )
         endfor

      endfor
   endif
endwhile
```

Figura B.5: Pseudo-Código RBLP.