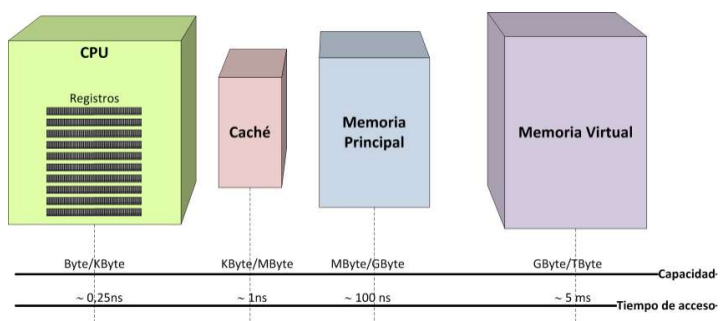
[Resumen clase 7](https://drive.google.com/file/d/1IUcb6sb3DZpZbVoB4uT9mSfIFoSQv7ju/view?usp=sharing)

# Sistema de memoria

Los programadores desean acceder a cantidades ilimitadas de memoria rápida. Surge la **jerarquía de memoria**

* Organizada en niveles que son ubicados en distintos lugares físicos
* Fabricados con tecnologías diferentes que se gestionan de manera independiente



* Objetivo: la velocidad del sistema deberá ser, aproximadamente, la del nivel más rápido al costo del nivel mas barato
* A medida que nos alejamos de la CPU, cada nivel inferior es más grande, más lento y más barato que el nivel superior en la jerarquía
* Debe haber correspondencia de direcciones en los distintos niveles

Propiedades a cumplir:

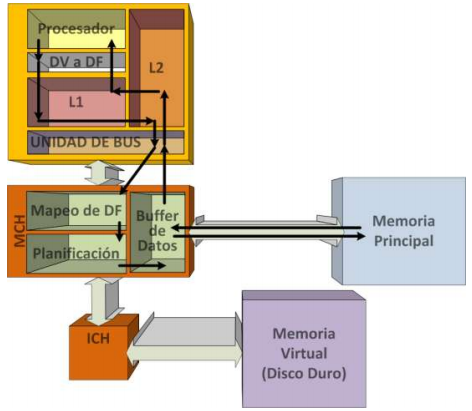
* Inclusión: Los datos almacenados en un nivel han de estar almacenados en los niveles inferiores a él
* Coherencia: Las copias de la misma información en los distintos niveles deben contener los mismos valores

**¿Por qué funciona la jerarquía?**

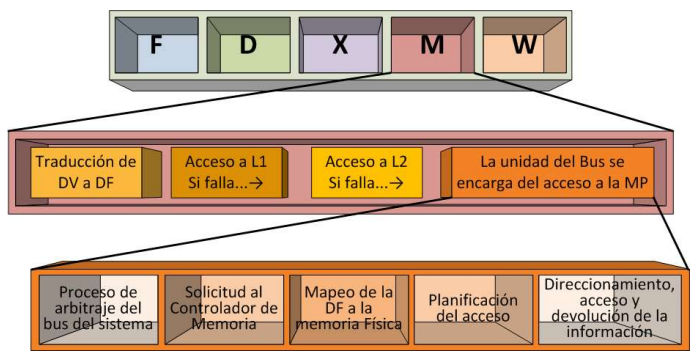
Por el principio de localidad de referencias

* Localidad temporal: los elementos de memoria referenciados recientemente (datos o instrucciones), volverán a serlo en un futuro próximo → subo la palabra de nivel
* Localidad espacial: Los elementos de memoria cuyas direcciones están próximas a los últimos referenciados serán referenciados → subo un bloque (con la palabra) de nivel

## Mecanismo de acceso a memoria

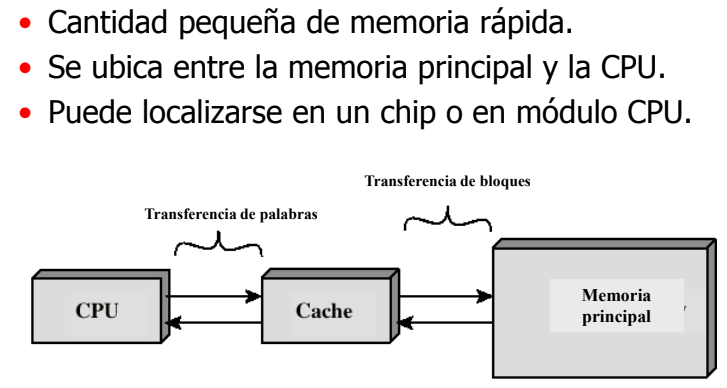


## Trabajo de etapa M del cauce

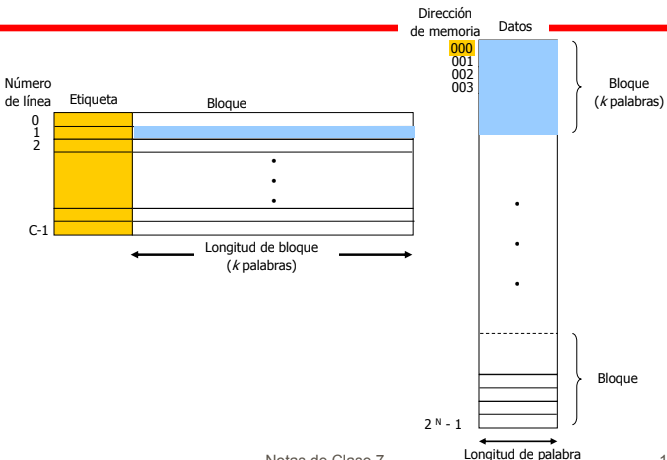


## Memoria cache

* Cantidad pequeña de memoria rápida
* Se ubica entre la memoria principal y la CPU
* Puede localizarse en un chip o en módulo CPU



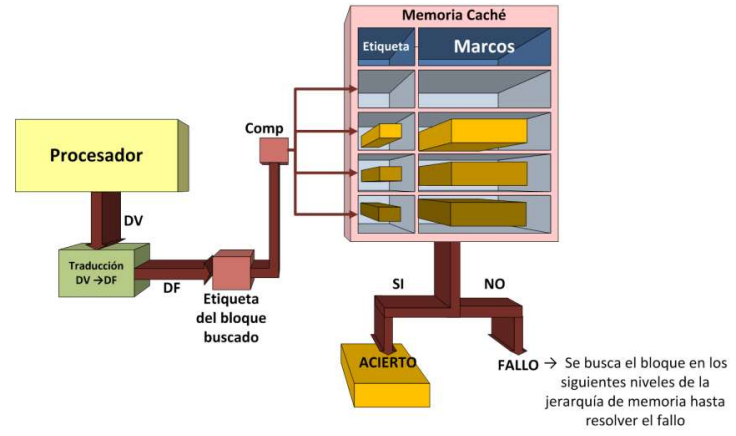
## Memoria caché y principal



## Funcionamiento de la caché

* La CPU solicita contenido de 1 dirección de memoria
* Si la caché tiene el dato, lo obtiene rápidamente de allí
* Si la caché no tiene el dato, se lee el bloque que contiene esa dirección desde la memoria principal y copia en la caché
  + Después, la caché entrega el dato requerido a la CPU

La caché incluye etiquetas para identificar qué bloque de la memoria principal está en cada una de sus líneas



## Conceptos básicos

* Acierto (hit): se encuentra en la caché el dato solicitado
* Fallo (miss): no se encuentra en la caché el dato solicitado
* Un bloque que contiene la palabra accedida se copia de la memoria principal a una línea de caché
* Tiempo para servir un fallo: depende de la latencia y ancho de banda de la memoria principal
  + Latencia: tiempo necesario para completar un acceso a memoria
  + Ancho de banda: cantidad de información por unidad de tiempo que puede transferirse desde/hacia la memoria
* Los fallos de caché se gestionan mediante hardware y causan que el procesador se detenga hasta que el dato esté disponible

## Prestaciones de la jerarquía

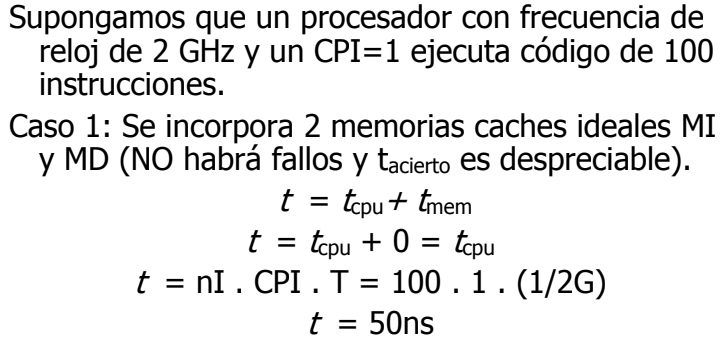
Tiempo de acceso medio a memoria

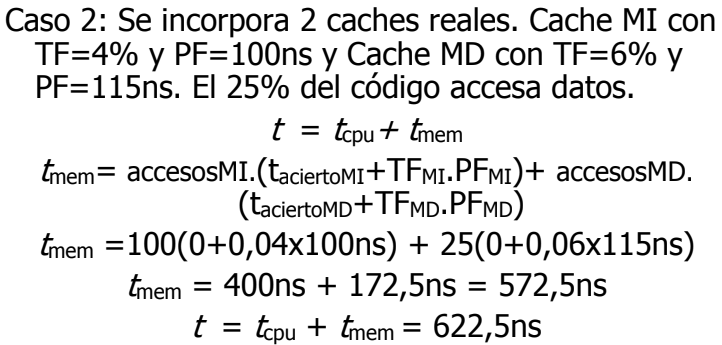
* Tacceso = Tacierto + Tfallos\_memoria
* Tfallos\_memoria = Tasa de fallos x Penalización\_fallo
* **Tacceso = Tacierto + TFxPF**

Para mejorar las prestaciones

* Reducir el tiempo en caso de acierto (Tacierto)
* Reducir la tasa de fallos (TF)
* Reducir la penalización por fallo (PF)

## Jerarquía perfecta vs real



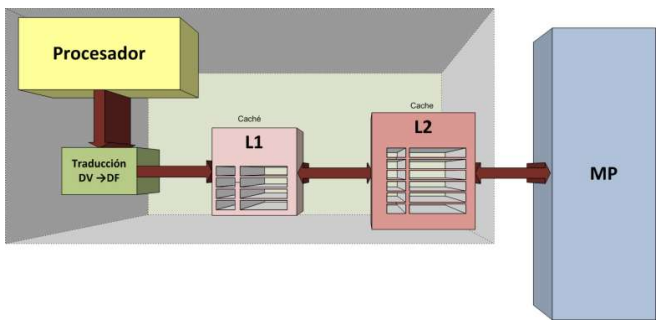


## Diseño de la caché

* Organización (tamaño y cantidad)
* Política de ubicación: tipo de función de correspondencia
* Política de reemplazo: algoritmo de sustitución
* Política de escritura

## Optimización de la caché

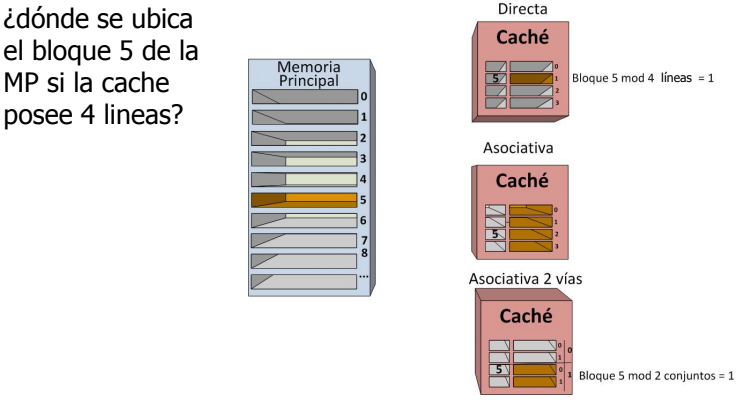
Tamaño-costo-niveles



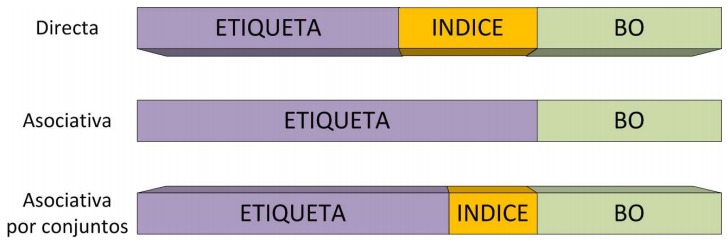
## Ubicación de un bloque

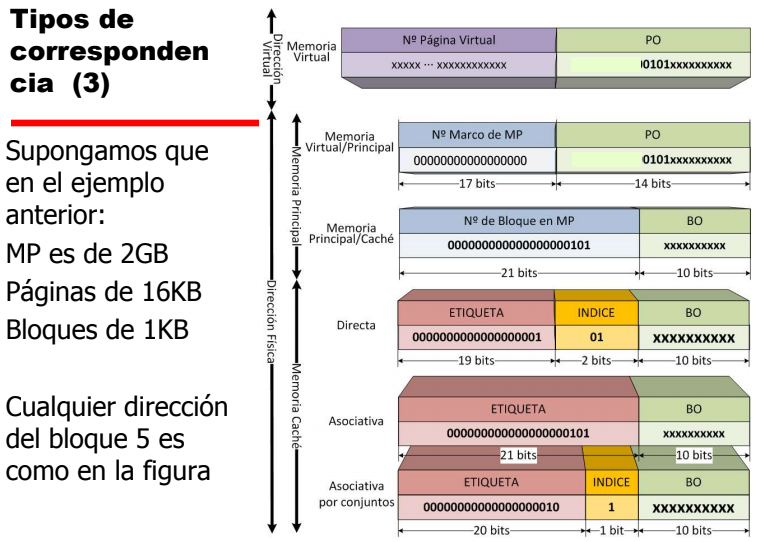
* Correspondencia directa. Un bloque solo puede estar almacenado en un lugar de la caché
  + N° línea caché = N° bloque ref. mod N° líneas caché
* Correspondencia total asociativa. Un bloque puede almacenarse en cualquier lugar de la caché
* Correspondencia asociativa por conjuntos. Un bloque puede almacenarse en un conjunto restringido de lugares en la caché.
  + Un conjunto es un grupo de líneas de la caché
  + N° conjunto = N° bloque ref. mod N° conjuntos caché

## Tipos de correspondencia

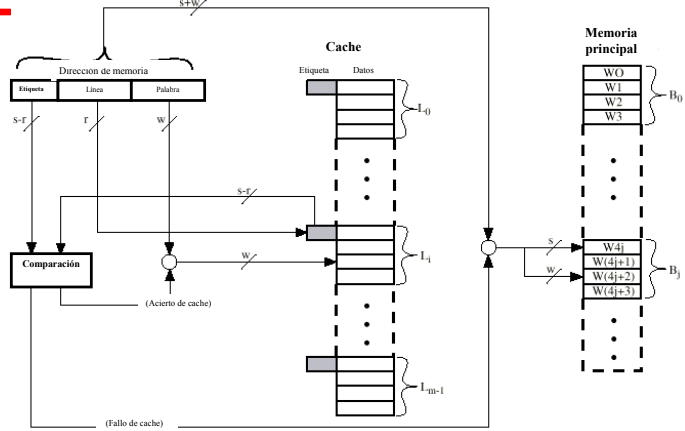


La interpretación de la dirección física depende del tipo que se utilice. INDICE indicará la línea ó el conjunto que le corresponde. BO representa todas las direcciones que pertenecen al bloque.

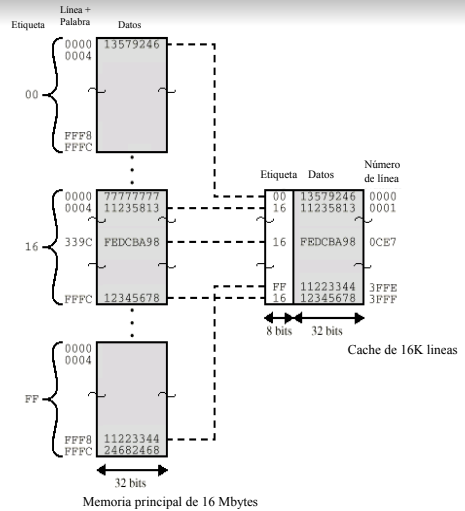




### Correspondencia directa: organización de la caché



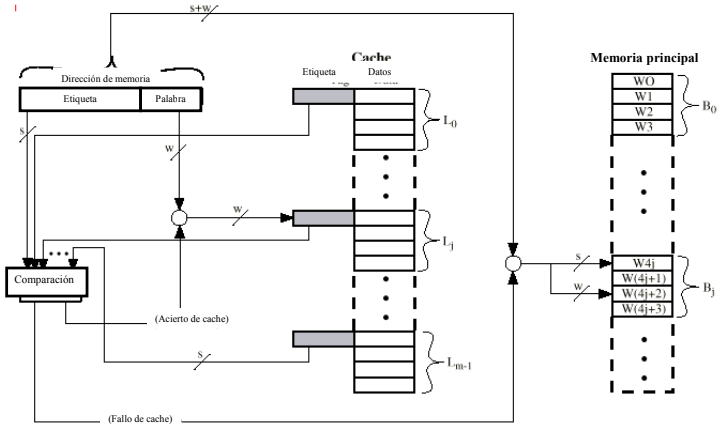
Ejemplo de correspondencia directa



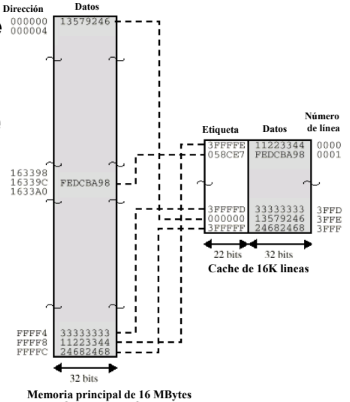
### Correspondencia directa: ventajas y desventajas

* Simple
* Poco costosa
* Hay una posición concreta para cada bloque dado: si un programa accede a dos bloques que se corresponden a la misma línea (diferentes bloques de memoria principal) de forma repetida, las pérdidas de caché (desaciertos) serán muy grandes

### Organización de caché totalmente asociativa



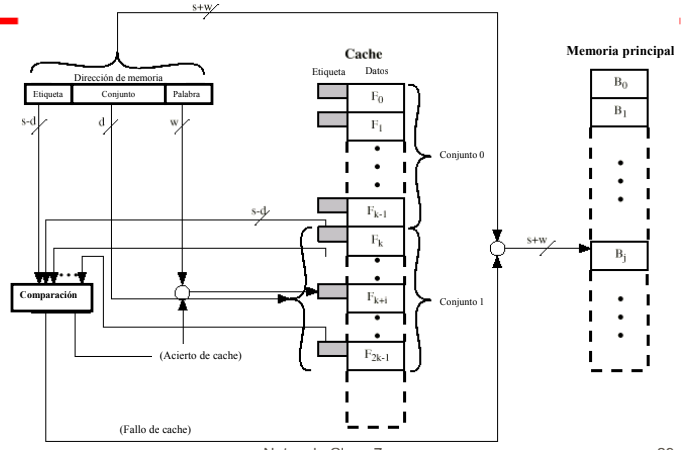
### Ejemplo de correspondencia totalmente asociativa



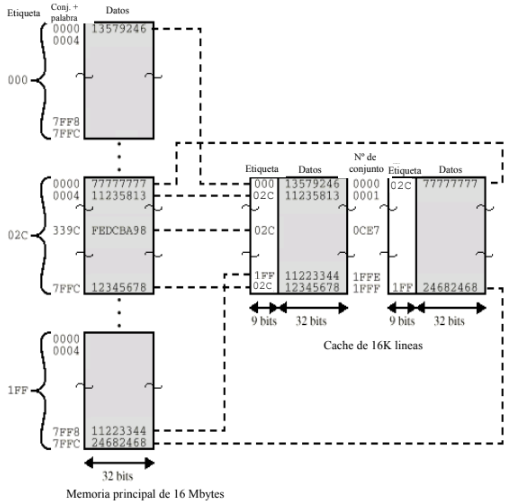
### Correspondencia asociativa: ventajas y desventajas

* Un bloque de memoria principal puede colocarse en cualquier línea de la cache
* La etiqueta identifica unívocamente un bloque de memoria
* Todas las etiquetas de las líneas se examinan para buscar una coincidencia
* Búsqueda costosa (en tiempo principalmente)

### Organización de caché asociativa por conjuntos



### Ejemplo de correspondencia asociativa por conjuntos de 2 vías



### Correspondencia asociativa por conjuntos: ventajas y desventajas

* Combina lo mejor de las otras correspondencias
* La caché se divide en un grupo de conjuntos: cada conjunto contiene un número de líneas.
  + N vías, con N=2,4,8,...etc.
* Un bloque determinado corresponderá a cualquier línea de un conjunto determinado
  + El bloque B puede asignarse en cualquiera de las líneas del conjunto i

## Política de reemplazos

Algoritmos de sustitución

* En correspondencia directa
  + El que ocupa el lugar del nuevo
* En correspondencia asociativa
  + LRU (menos recientemente usado)
  + FIFO (más antiguo)
  + LFU (menos frecuentemente usado)
  + Aleatoria

### Algoritmos de sustitución

* No hay elección
* Sólo hay una posible línea para cada bloque
* Se necesita una sustitución de esa línea sí o sí

Correspondencias asociativas

* Los algoritmos deben implementarse en hardware (para conseguir velocidad)
* Menos recientemente usado (LRU)
  + Requiere controles de tiempos
  + En correspondencias asociativas por conjuntos de 2 vías. Cuál de las 2 líneas es la LRU?
* Primero en entrar-primero en salir (FIFO)
  + Requiere controles de acceso
  + Se sustituye aquella línea que ha estado más tiempo en la caché
* Menos frecuentemente usado (LFU)
  + Requiere controles de uso
  + Se sustituye aquella línea que ha experimentado menos referencias
* Aleatoria
  + Se sustituye una línea al azar

## Política de escritura

Se debe evitar la inconsistencia de memorias en el caso de escrituras.

Tener en cuenta:

* La CPU escribe sobre una línea de caché: el bloque de memoria principal correspondiente debe ser actualizado en algún momento.
* Un módulo E/S puede tener acceso directo a la memoria principal
* En procesamiento paralelo, las múltiples CPU pueden tener caches individuales

### Políticas de escritura: en acierto

Write-through (escritura inmediata)

* Se actualizan simultáneamente la posición de la caché y de la memoria principal
  + Con múltiples CPU, observar el tráfico a memoria principal para mantener actualizada cada cache local
  + Se genera mucho tráfico y retrasa la escritura

Write-back (post-escritura)

* La información sólo se actualiza en la caché
  + Se marca como actualizada → bit de “sucio”
  + La memoria principal se actualiza en el reemplazo y puede contener información errónea en algún momento

### Políticas de escritura: en fallo

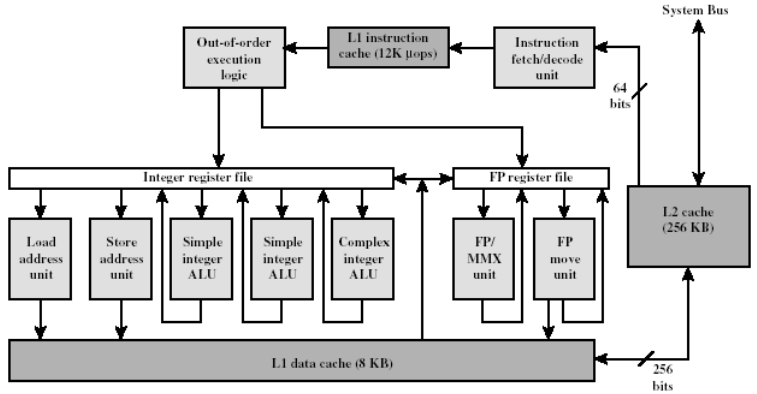
Write allocate

* La información se lleva de la memoria principal a la caché. Se sobreescribe en la caché
  + Habirtual con write-back

No-write allocate

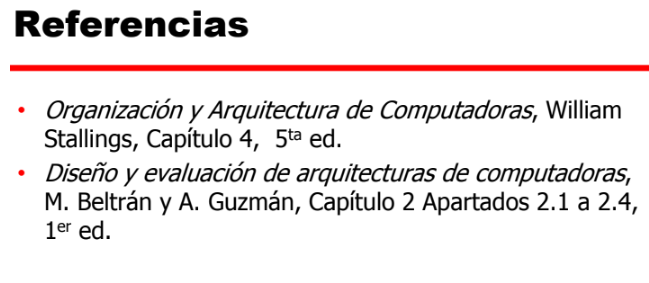
* El bloque no se lleva a la memoria caché. Se escribe directamente en la memoria principal
  + Habitual con write-through

### Pentium 4



Puede tener hasta 3 niveles de caché:

* Caches L1 separadas para datos e instrucciones
  + Cache de datos (de 8 KBytes). Asociación por conjuntos de 4 vías. Bloques de 64 bytes. Política de Escritura inmediata. Acceso a los datos enteros en dos ciclos de reloj.
  + La caché de instrucciones almacena segmentos de caminos de ejecución de instrucciones decodificadas (trazas)
* Cache L2 (interna) unificada para datos e instrucciones
  + Capacidad de 256 KBytes. Organización asociativa por conjuntos de 8 vias. Bloques de 128 bytes. Política de Post-escritura. Latencia de acceso de 7 ciclos de reloj.
* Las dos caches (L1 y L2) en el chip del procesador. Ancho de banda de las transferencias entre L1 y L2 48 Gbytes/s
* La arquitectura admite un tercer nivel de caché (L3) en el mismo chip (servidores)



[Anexo clase 7](https://drive.google.com/file/d/1viN8xN79Alf5nY0HHASoTCWcXl1GGq52/view?usp=sharing)