**Patrones**

* El resolver problemas efectivamente involucra mas que solo encontrar una solución. Una vez se encuentre la solución, se debe invertir esfuerzo en su mejora y optimización, para ello el reconocimiento y exploración de patrones.
* Cuando se analiza un problema y se empieza a trabajar en la solución probablemente se notará que algunos elementos se repiten o son muy similares a otros. Si no se identifican patrones en una solución se recomienda realizar un esfuerzo para buscarlos, poque son el primer paso para una solución mas manejable y poderosa.

**Patrones simples**

* Para poder identificar o reconocer patrones simples debemos de buscar sustantivos que se repitan constantemente, pueden corresponde a objetos de la solución. Buscar verbos que aparezcan repetidamente. Pueden ser operaciones que lleve a cabo la solución. Buscar descripciones concretas, como adjetivos indican propiedades de las cosas.

**Patrones complejos**

* Para identificar patrones mas complejos se requiere ampliar el alcance, considerando grupos de instrucciones a la vez. Requiere experiencia, pero vale la pena.
* Debemos de buscar patrones en secuencias de instrucciones u operaciones pueden ser generalizados en ciclos. Patrones entre grupos separados de instrucciones pueden ser generalizados en subrutinas. Patrones dentro de condicionales o ecuaciones pueden ser generalizados en reglas.

**Ciclos**

* Son una serie de pasos que se ejecutan repetidamente. Es un tipo de generalización. Se deben observar las formas especificas y luego derivar a una mas general. Un bloque de instrucciones muy repetitivos puede facilitar la identificación. Por ejemplos considere este bloque.
* Al menos dos pasos se repiten igual que los primeros dos. Se podría colocar en un ciclo de esta manera. Establecer coordenadas.

**Subrutinas**

* Bloques separados de instrucciones. Se pueden declarar los pasos en una subrutina que puede ser invocada desde diferentes lugares y este empaquetada en una unidad.

**Reglas**

* Se pueden declarar estos pasos en una subrutina que pueda ser invocada desde diferentes lugares y esta empaquetada en una unidad.