|  |
| --- |
| TAD Pilas |
| Pilas= ((e1, e2, e3, ..., en), top) |
| {inv: 0 <- n /\ Size (Stack) = n /\ top =en |
| isEmpty: ->Pila  size: pilas x elemento -> pilas  pop: pilas -> pilas  Top: pilas  Push: pilas  Barajar: pilas ->elemento |
| isEmpty ()    “Este método verifica si la estructura de datos está vacía”   {pre: N/A}    {pos: Pila s = ∅ } |
| Push ()    “Agregar un elemento e a la pila s”    {pre: pilas s = (e1, e2, e3, ..., en) e=elemento, s = ∅ }    {pos: pilas s = (e1, e2, e3, ..., en, e), s = (e)} |
| pop ()    “sacar el último elemento agregado”    {pre: pila s=(e1,e2,e3……..en)}    {pos: pilas s = (e1, e2, e3, ..., en-1)} |
| top ()    “se recibe el valor del elemento que este arriba de la pila”    {pre: pila s=(e1,e2,e3……..en)}    {pos: Elemento en} |
| size ()    “Devuelve el número de elementos en la pila”    {pre: pila s}      {pos: Devuelve el tamaño actual de la pila.} |
| Barajar ()    “Mezcla las cartas en el mazo.”    {pre: N/A}    {pos: Las cartas en el mazo se mezclan aleatoriamente, cambiando su orden.} |
| crearMazo()  “Este método se encarga de crear un mazo de cartas para el juego.” |
| {pre: N/A}  {pos: El mazo de cartas (this.cartas) estará completo y correctamente generado, listo para ser utilizado en el juego.} |
| robarCarta()  “Este método permite a un jugador robar una carta del mazo” |
| {pre: N/A}  {pos: El mazo de cartas (this.cartas) contendrá una carta menos que antes de llamar a este método}  mezclar()  “Este método mezcla las cartas en el mazo para garantizar un orden aleatorio antes de comenzar el juego.”  {pre: N/A}  {pos: El mazo de cartas (this.cartas) estará completamente mezclado, asegurando un orden aleatorio para las cartas} |
|  |