tint compare To(String s)  int compare To(String s)  String concat(String s)  Devuelve true si la cadena que lama al método, pero sin que se squal al array de caracteres data.  Devuelve true si la cadena termina con el texto s.  Compara ambas cadenas, devuelve true si on iguales, sino false.  Compara ambas cadenas, devuelve true si on iguales, sino false.  Compara ambas cadenas como equals sin tener en cuenta la minuscola si sin discola si sin discola si sin discola si sin discola si sin tener en cuenta la minuscola si sin discola si sin sin discola si si posicion si sin discola si	Result.	Método	Descripción
int compareTo(String s)  Compara is dos cadenas affabeticamente (ASCII). Devuelve un valor menor que cero si la cadena que invoca el método es menor que se, devuelve o si son iguales y devuelve un valor meyor que cero si la cadena que invoca al método es menor que se, devuelve o si son iguales y devuelve un valor meyor que s.  String concat(String s)  String concat(String s)  Añade la cadena que invoca al método es menor que se, devuelve o si monitor dos cadenas que lama al método, pero sin que se actualice la original.  String (static)  String. String.copyValueOf(char[] data)  boulean cadsWith(String s)  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  String (static)  String, format(String s)  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Devuelve un string com el formato false or fel/los objetos o  Tiene que haber el mismo número de argumentos que elementos estén invoca.  Pevuelve un array de caracteres que toma a partir de la cadena que lo invoca estén inderedado per lo invoca se desenve que true si a cadena en el array de caracteres desta destina de la posición de faltes o en la cadena que lo invoca en la cadena de la cadena que lo invoca en la cadena de la cadena que lo invoca em la cadena de la cadena que lo invoca em la cadena de la cadena que lo inv	char	charAt(int index)	
int compareTo(String s)  int compareTo(sproreCase(String s)  int compareTo[sproreCase(String s)  String concat(String s)  String concat(String s)  String String string (chatic)  String concat(String s)  String concat(String s)  String concat(String s)  String string (chatic)  String concat(String s)  String concat(String s)  String concat(String s)  String string (chatic)  String compareTo(sproreCase(String s)  String string (chatic)  String compareTo(sproreCase(String s)  Devuelve true si la cadena termina con el texto s.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas como equals sin tener en cuenta las mayúsculas y las minúsculas.  String (static)  String,format(String f, Objeto o,)  String,format(String f, Objeto o,)  String getBytes()  Devuelve un string con el formato figo opr f de/Ros objeto/s o.  Tiene que haber el mismo número de argumentos que elementos estén referidos en el formato.  Devuelve un string con el formato figo opr f de/Ros objeto/s o.  Tiene que haber el mismo número de argumentos que elementos estén referidos en el formato.  Devuelve un string con el formato figo opr f de/Ros objeto/s o.  Tiene que haber el mismo número de argumentos que elementos estén el mosición en formato.  Devuelve un string con el formato figo opr f de/Ros objeto/s o.  Tiene que haber el mismo número de argumentos que elementos estén el mosición el formato.  Devuelve un string con el formato figo oprimero de argumentos que elementos estén el mosición el formato.  Devuelve la posición del texto se en la cadena que lo invoca.  Tiene que haber el mismo número de argumentos que elemento de la cadena que lo invoca en la cadena que lo invoca e	Citat	chai / t(int index)	
devuelve 0 si son iguales y devuelve un valor mayor que cero si la cadena que invoca al método es mayor que son si la cadena que invoca al método es mayor que son si la cadena que invoca al método es mayor que son si menta si el texto es mayúsculas o no.  String concat(String s)  String String, String, String, String, String, String string endate in cadena que inma al método, pero sin que se actualide la original.  Produce un objeto String que es igual al array de caracteres data.  Devuelve true si la cadena atermina con el texto s.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  String getBytes()  Devuelve un array de caracteres data.  Devuelve un array de caracteres dest.  Toma los caracteres deste al posición sordegin hasta la posición sercin y los copia en el array desde la posición distregin.  Devuelve la posición del texto s en la cadena que lo invoca.  Devuelve la posición del texto s en la cadena que lo invoca en cuentra devuelve -1.  Como indexOf(String s, int primeraPos)  int length()  Devuelve true solo si la longitud de la cadena que lo invoca en cadena que lo invoca, pero empezando a buscar desde la posición del texto s en la cadena que lo invoca.  Devuelve la última posición en que aparece la cadena del texto sen la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la longitud de la cadena que lo inv	int	compareTo(String s)	, ,
int compareToIgnoreCase(String s)  Compare Godena que invoca al metodo es mayor que cero si se cadena que invoca al metodo es mayor que cero si se cadena que invoca al metodo es mayor que so si cadena que invoca al metodo es mayor que so si cadena que invoca al metodo es mayor que so si cadena que ilama al método, pero sin que se sextualice la original.  String concat(String s)  String String.CopyValueOf(char[] data)  boolean equals(String s)  Devuelve true si la cadena que ilama al método, pero sin que se sigual al array de caracteres data.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Devuelve un string con el formato.  Devuelve un string con el formato.  Devuelve un array de caracteres que temorio este nreferidos en el formato.  Como indexOf (String s)  Devuelve un array de caracteres dest.  Toma los caracteres desde la posición streguin hasta la posición sur devuelve a posición del texto sen la cadena que lo invoca.  Siring replace(far carAnterior, char carNuevo)  Devuelve la última posición en que aparece la cadena del texto sen la cadena que lo invoca.  Devuelve la última posición en que aparece la cadena que lo invoca.  Devuelve la última posición en que aparece la cadena que lo invoca.  Devuelve la última posición en que aparece la cadena que lo invoca.  Devuelve la última posición en que aparece la cadena que lo invoca.  Cambia la tod			
tint compareToIgnoreCase(String s)  String concat(String s)  String copyValueOf(char   data)  Produce un objeto String que es igual al array de caracteres data.  Devuelve true si la cadena termina con el texto s.  Devuelve true si la cadena termina con el texto s.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  String (static)  String, format(String s)  String, format(String f, Objeto o,)  Betaltic string format(String f, Objeto o,)  String (static)  byte    getBytes(in srcBegin, int srcEnd, char   dest, int dstBegin);  int indexOf(String s)  Devuelve un array de caracteres due la posición srcBegin hasta la posición srcBegin del texto sen la cadena que lo invoca.  Almacena el contenido de la cadena en el array de caracteres dest. Toma los caracteres desde la posición srcBegin hasta la posición primera?  Int indexOf(String s, int primera?os)  Devuelve la posición del texto sen la cadena que lo invoca.  String l'astalndexOf(String s, int primera?os)  Devuelve la bosición del texto sen la cadena que lo invoca en lo cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca en lo cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca en la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Cadena q			
String  concat(String s)  String copy ValueO((char   data)  featible  catalogy and concat(String s)  String copy ValueO((char   data)  catalogy and concat(String s)  String copy ValueO((char   data)  catalogy and catalogy and concates a final de la cadena que llama al método, pero sin que se actualice la original.  String copy ValueO((char   data)  catalogy and catal			
String concat(String s)  String concat(String s)  String string  String copyValueOf(char[ data)  condat(String s)  String string  String string  String copyValueOf(char[ data)  condat(String s)  Devuelve true si la cadena a termina con el texto s.  Devuelve true si la cadena a termina con el texto s.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Devuelve un array de caracteres de true si son iguales, sino false.  Devuelve un array de caracteres de true si son iguales, sino false.  Compara ambas cadenas, devuelve true si son iguales, sino false.  Devuelve un array desce la gosición siguales providos de la cadena que lo invoca.  Com indexOf(String s)  Devuelve la opsición del texto sen la cadena que lo invoca cadena que lo invoca.  Com indexOf(String s)  Devuelve la iditina posición del texto sen la cadena que lo invoca.  Com indexOf(String s)  Devuelve la iditina posición del texto sen la ca	int	compareToIgnoreCase(String s)	
String   Concat(estring s)   que se actualice la original.		1 8 (	·
String   String.copyValueOf(char   data)   Produce un objeto String que es igual al array de caracteres data   Devuelve true si la cadena termina con el texto s.	String	concat(String s)	· · · · · · · · · · · · · · · · · · ·
String   Strings opt value (Italian   Italian   Italia	Ü	, ,	que se actualice la original.
Devuelve true si la cadena termina con el texto s.		String.copyValueOf(char[] data)	Produce un objeto String que es igual al array de caracteres data.
Compara ambas cadenas, devuelve true si son iguales, sino false.	`	andsWith(Strings)	Downship two side and an atomina can altoute s
Compara ambas cadenas como equals sin tener en cuenta las mayúsculas y las minúsculas.			
String (static)   String.format(String f, Objeto o,)   Devuelve un String con el formato fijado por f de/los objeto/s o. Tiene que haber el mismo número de argumentos que elementos estén referidos en el formato.	boolean	equals(String s)	
String (static)   String.format(String f, Objeto o,)   Devuelve un String con el formato fijado por f de/los objeto/s o.   Tiene que haber el mismo número de argumentos que elementos estén referidos en el formato.   Devuelve un array de caracteres que toma a partir de la cadena que lo invoca.   Devuelve un array de caracteres dest.   Toma los caracteres desde la posición scrBegin hasta la posición into indexOf(String s)   Devuelve la posición del texto s en la cadena que lo invoca.   Si no lo encuentra devuelve 1.   Como indexOf anterior, devuelve la posición del texto s en la cadena que lo invoca esto ne primera pos into lastIndexOf(String s, int primeraPos)   Devuelve la indexOf anterior, devuelve la posición del texto s en la cadena que lo invoca esto ne que aparece la cadena que lo invoca esto ne que aparece la cadena que lo invoca esto ne que aparece la cadena que lo invoca esto ne la cadena que lo invoca esto ne que parece la cadena que lo invoca esto ne la cadena que lo invoca empezando a buscar desde la posición primeraPos.	boolean	equalsIgnoreCase(String s)	· · · · · · · · · · · · · · · · · · ·
String (static)   String,format(String f, Objeto o,)   Tiene que haber el mismo número de argumentos que elementos estén referidos en el formato.			
estén referidos en el formato.  Devuelve un array de caracteres que toma a partir de la cadena que lo invoca. Almacena el contenido de la cadena en el array de caracteres deste la posición srcEndy hasta la posición srcEndy los copia en el array de caracteres deste la posición srcEndy los copia en el array de caracteres deste la posición srcEndy los copia en el array de caracteres deste la posición del texto sen la cadena que lo invoca. Si no lo encuentra devuelve -1.  Como indexOf (String s, int primeraPos)  boolean isEmpty() Devuelve true solo si la longitud de la cadena que lo invoca es o.  Devuelve la última posición del texto sen la cadena que lo invoca es o.  Devuelve la última posición del texto sen la cadena que lo invoca.  Devuelve la última posición del texto sen la cadena que lo invoca.  Beruelve la última posición del texto sen la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca es o.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  String replace(char carAnterior, char carNuevo)  Teplace(char carAnterior,		String format(String f Ohieto o )	
Devuelve un array de caracteres que toma a partir de la cadena que lo invoca.	(static)	String. for mat(String 1, Objeto 0,)	,
getBytes(int srcBegin, int srcEnd, char]   dest, int distBegin);			
void   getBytes(int srcBegin, int srcEnd, char]   dest, int dstBegin);	byte[]	getBytes()	
Toma los caracteres desde la posición srcBegin hasta la posición del texto se na cadena que lo invoca. Si no lo encuentra devuelve -1.  Int indexOf(String s) Devuelve la posición del texto se na cadena que lo invoca pero empezando a buscar desde la posición del texto se na cadena que lo invoca, pero empezando a buscar desde la posición primeraPos.  Int lastIndexOf(String s) Devuelve true solo si la longitud de la cadena que lo invoca se 0.  Devuelve la última posición en que aparece la cadena del texto se na cadena que lo invoca.  Devuelve la última posición en que aparece la cadena del texto se na cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Int length() Devuelve la iditima posición del texto se na cadena que lo invoca empezando a buscar desde la posición primeraPos.  Devuelve la última posición en que aparece la cadena del texto se na cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve true si la cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve true si la cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve una cadena idéntica a la original en la que ha cambiando los caracteres iguales a caranterior por carn\uevo. La cadena original no se modifica. También produce el cambio si los parámetros son una secuencia de caracteres.  String replaceFirst(String regex, String str2) Cambia la todas las apariciones de la de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no cambia.  Cambia la primera paración de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.  String toCharArray() Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve true si la cadena a minúsculas. La original no se modifica.  String toLowerCase() Convierte la cadena a minúsculas. La original no se modifica.  Elimina los blancos			
int indexOf(String s)  Devuelve la posición del texto s en la cadena que lo invoca. Si no lo encuentra develvel - 1.  Como indexOf anterior, devuelve la posición del texto s en la cadena que lo invoca. Si no lo encuentra develvel - 1.  Como indexOf anterior, devuelve la posición del texto s en la cadena que lo invoca, pero empezando a buscar desde la posición primeraPos.  Devuelve true solo si la longitud de la cadena que lo invoca es 0.  Devuelve la última posición en que aparece la cadena del texto s en la cadena que lo invoca.  Devuelve la última posición en que aparece la cadena del texto s en la cadena que lo invoca.  Devuelve la última posición en que aparece la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la última posición del texto s en la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la última posición en que aparece la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la última posición en que aparece la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la última posición en que aparece la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la longitud de la cadena que lo invoca.  Devuelve true si la cadena que lo invoca cumple la expresión regular, false si no la cumple la expresión regular, false si no la cumple la expresión regular, false si no la cumple la expresión regular regax (puede ser una cadena) por la cadena str2. La original no sem difica.  String replaceFirst(String regex, String str2)  Devuelve una cadena) por la cadena str2. La original no cambia.  Cambia la primera aparición de la expresión regular regax (puede ser una cadena) por la cadena str2. La original no sem difica.  Cambia la primera aparición de la expresión regular regax (puede ser una cadena) por la cadena str2. La original no sem difica.  String toCharArray()  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve true si la cadena que lo inv	void		·
Devuelve la posición del texto s en la cadena que lo invoca. Si no lo encuentra devuelve -1.   Int	VOIG	dest, int dstBegin);	· · · · · · · · · · · · · · · · · · ·
int indexOf(String s, int primeraPos)  boolean isEmpty()  boolean isEmpty()  boolean isEmpty()  boolean int lastIndexOf(String s)  int lastIndexOf(String s)  boolean int lastIndexOf(String s)  int lastIndexOf(String s)  int lastIndexOf(String s, int primeraPos)  boolean int lastIndexOf(String s)  int lastIndexOf(String s, int primeraPos)  int length()  boolean match(String regex)  boolean match(String regex)  boolean match(String regex)  creplace(char carAnterior, char carNuevo)  String replace(char carAnterior, char carNuevo)  String replace(string regex, String str2)  String replaceFirst(String regex, String str2)  String replaceFirst(String regex, String str2)  boolean startsWith(String s)  cadena que lo invoca cample la expresión regular, false si no la cumple  Devuelve true si la cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve una cadena identica a la original en la que ha cambiando los caracteres iguales a carAnterior por carNuevo. La cadena original no se modifica. También produce el cambio si los parámetros son una secuencia de caracteres.  Cambia la todas las apariciones de la de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no cambia.  Cambia la primera aparición de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.  String substring(int primeraPos, intsegundaPos)  char[  toCharArray() Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve una cadena) por la cadena str2. La original no se modifica.  String toLowerCase() Convierte la cadena a minúsculas. La original no se modifica.  String trim() Convierte la cadena a mayúsculas. La original no se modifica.  String trim() Selementa a mayúsculas. La original no se modifica.  String trim() Selementa cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.			
int indexOf(String s, int primeraPos)  boolean isEmpty()  boolean isEmpty()  int lastIndexOf(String s)  int length()  boolean match(String s, int primeraPos)  boolean match(String s, int primeraPos)  String replace(char carAnterior, char carNuevo)  String replaceAll(String regex, String str2)  String replaceAll(String regex, String str2)  String substring(int primeraPos, intsegundaPos)  String substring(int primeraPos, intsegundaPos)  String totCharArray()  String trin()  String split(String regex)  Como indexOf anterior, devuelve la posición del texto s en la cadena que lo invoca es o. Devuelve la última posición en que aparece la cadena del texto s en la cadena que lo invoca.  Devuelve la última posición del texto s en la cadena que lo invoca.  Devuelve la última posición del texto s en la cadena que lo invoca.  Devuelve la última posición del texto s en la cadena que lo invoca.  Devuelve la última posición del texto s en la cadena que lo invoca.  Devuelve true si la cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve una cadena identica a la original no la cadena de caracteres.  Cambia la todas las apariciones de la de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no cambia.  Cambia la primera aparición de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.  String substring(int primeraPos, intsegundaPos)  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve el texto que va desde primeraPos a segundaPos. Empieza en 0 y segundaPos no entra.  Convierte la cadena a manúsculas. La original no se modifica.  String trim()  Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.  Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.	int	indexOf(String s)	·
int indexOf(String s, int primeraPos)  boolean isEmpty()  Devuelve true solo si la longitud de la cadena que lo invoca es 0.  Devuelve la última posición en que aparece la cadena del texto s en la cadena que lo invoca.  Devuelve la última posición en que aparece la cadena del texto s en la cadena que lo invoca.  Devuelve la última posición del texto s en la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la última posición del texto s en la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la última posición del texto s en la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la última posición del texto s en la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la última posición del texto s en la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la última posición en que aparece la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve true si la cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve una cadena idéntica a la original en la que ha cambiando los caracteres iguales a carAnterior por carNuevo. La cadena original no se modifica.  String replace(char carAnterior, char carNuevo)  Devuelve una cadena jor la cadena str2. La original no se modifica.  Cambia la primera aparición de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.  String substring(int primeraPos, intsegundaPos)  Devuelve una cadena que lo invoca comienza con el texto s.  Devuelve el texto que va desde primeraPos a segundaPos. Empieza en 0 y segundaPos no entra.  Convierte la cadena a minúsculas. La original no se modifica.  String trim()  Devuelve una cadena que lo invoca. Los trozos s			
Devuelve true solo si la longitud de la cadena que lo invoca es 0.	int	indexOf(String s int primeraPos)	· ·
int lastIndexOf(String s)  Devuelve la última posición en que aparece la cadena due lo invoca es 0.  Devuelve la última posición en que aparece la cadena del texto s en la cadena que lo invoca.  Devuelve la última posición en que aparece la cadena due lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la última posición de letexto s en la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la última posición de letexto s en la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la última posición de la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la última posición de la cadena que lo invoca.  Devuelve la última posición de la cadena que lo invoca.  Devuelve true si la cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve una cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve una cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve una cadena que lo invoca cumple la expresión regular regex (puede ser una cadena) por la cadena str2. La original no cambia.  Cambia la todas las apariciones de la de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.  String substring(int primeraPos, intsegundaPos)  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve el texto que va desde primeraPos a segundaPos. Empieza en 0 y segundaPos no entra.  Charl la toCharArray()  Devuelve un array de caracteres a partir de la cadena dada.  String toLowerCase()  Convierte la cadena a minúsculas. La original no se modifica.  String trim()  Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.	1110	indexor(sering 5, me printerar 65)	
int lastIndexOf(String s)  Devuelve la última posición en que aparece la cadena del texto s en la cadena que lo invoca.  Devuelve la última posición del texto s en la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Int length()  Devuelve la longitud de la cadena que lo invoca.  Devuelve true si la cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve una cadena idéntica a la original en la que ha cambiando los caracteres iguales a carAnterior por carNuevo. La cadena original no se modifica. Tambien produce el cambio si los parámetros son una secuencia de caracteres.  String replace(Istring regex, String str2)  String replaceFirst(String regex, String str2)  String substring(int primeraPos, intsegundaPos)  String toCharArray()  Devuelve true si la cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve una cadena idéntica a la original en la que ha cambiando los caracteres iguales a carAnterior por carNuevo. La cadena original no se modifica. Tambien produce el cambio si los parámetros son una secuencia de caracteres.  Cambia la todas las apariciones de la de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no cambia.  Cambia la primera aparición de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.  String substring(int primeraPos, intsegundaPos)  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve el texto que va desde primeraPos a segundaPos. Empieza en 0 y segundaPos no entra.  Convierte la cadena a minúsculas. La original no se modifica.  String toCharArray()  Devuelve un array de caracteres a partir de la cadena dada.  String toLowerCase()  Convierte la cadena a minúsculas. La original no se modifica.  Elimina los blancos que tenga la cadena tanto al principio delante como al final, sin modificar la cadena que lo invoca. Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la	hooloon	isEmpty()	· ·
int lastIndexOf(String s, int primeraPos)  int length()  Devuelve la última posición del texto s en la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la longitud de la cadena que lo invoca.  Devuelve la longitud de la cadena que lo invoca.  Devuelve true si la cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve una cadena idéntica a la original en la que ha cambiando los caracteres iguales a carAnterior por carNuevo. La cadena original no se modifica. También produce el cambio si los parámetros son una secuencia de caracteres.  String replaceAll(String regex, String str2)  String replaceFirst(String regex, String str2)  String substring(int primeraPos, intsegundaPos)  String substring(int primeraPos, intsegundaPos)  Cambia la primera aparición de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve una rray de caracteres a partir de la cadena dada.  String toUpperCase()  Convierte la cadena a mayúsculas. La original no se modifica.  String trim()  Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.  Povuelve la cadena que lo invoca.	Doolean		
int length()  Devuelve la última posición del texto s en la cadena que lo invoca, empezando a buscar desde la posición primeraPos.  Devuelve la longitud de la cadena que lo invoca.  Devuelve true si la cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve una cadena idéntica a la original en la que ha cambiando los caracteres iguales a carAnterior por carNuevo. La cadena original no se modifica. También produce el cambio si los parámetros son una secuencia de caracteres.  String replaceAll(String regex, String str2)  String replaceFirst(String regex, String str2)  Devuelve una cadena) por la cadena str2. La original no cambia.  Cambia la primera aparición de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.  String substring(int primeraPos, intsegundaPos)  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve un array de caracteres a partir de la cadena dada.  String toUpperCase()  Convierte la cadena a minúsculas. La original no se modifica.  String string()  String split(String regex)  String split(String regex)  Devuelve en array de caracteres a partir de la cadena dada.  Elimina los blancos que tenga la cadena tanto al principio delante como al final, sin modificar la cadena que lo invoca.  String split(String regex)  Devuelva cadena que lo invoca comienzo con el costerión de la expresión regular.	int	lastIndexOf(String s)	
int length()  Devuelve la longitud de la cadena que lo invoca.  Devuelve true si la cadena que lo invoca cumple la expresión regular, false si no la cumple  Devuelve una cadena identica a la original en la que ha cambiando los caracteres iguales a carAnterior por carNuevo. La cadena original no se modifica. También produce el cambio si los parámetros son una secuencia de caracteres.  String replaceAll(String regex, String str2)  String replaceFirst(String regex, String str2)  Devuelve una cadena) por la cadena str2. La original no cambia. Cambia la primera aparición de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.  String substring(int primeraPos, intsegundaPos)  String toCharArray()  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve el texto que va desde primeraPos a segundaPos. Empieza en 0 y segundaPos no entra.  Char[  toCharArray()  Devuelve un array de caracteres a partir de la cadena dada.  String toUpperCase()  Convierte la cadena a mayúsculas. La original no se modifica.  String trim()  String split(String regex)  Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra on el lengtanto se substrior de la expresión regular.			·
tring replace(char carAnterior, char carNuevo)  String replaceAll(String regex, String str2)  String replaceFirst(String regex, String str2)  String replaceFirst(String regex, String str2)  String substring(int primeraPos, intsegundaPos)  String toCharArray()  String toCharArray()  String toCharArray()  String	int	lastIndexOf(String s, int primeraPos)	
String replace(char carAnterior, char carNuevo)  String replace(char carAnterior, char carNuevo)  String replaceAll(String regex, String str2)  String replaceFirst(String regex, String str2)  String substring(int primeraPos, intsegundaPos)  String toCharArray()  String toCharArray()  String toCharArray()  String string toCharArray()  String string toCharArray()  String toLowerCase()  String toUpperCase()  String trin()  String split(String regex)  String split(String regex)  String toUpperCase()  String toUpperCase()  String toUpperCase()  String split(String regex)  String split(String	int	length()	
String replace(char carAnterior, char carNuevo)  String replace(char carAnterior, char carNuevo)  String replaceAll(String regex, String str2)  String replaceFirst(String regex, String str2)  String substring(int primeraPos, intsegundaPos)  String toCharArray()  String toCharArray()  String toCharArray()  String string toCharArray()  String string toCharArray()  String toLowerCase()  String toUpperCase()  String trin()  String split(String regex)  String split(String regex)  String toUpperCase()  String toUpperCase()  String toUpperCase()  String split(String regex)  String split(String		. 1.60.	Devuelve <b>true</b> si la cadena que lo invoca cumple la expresión regular,
Pevuelve una cadena idéntica a la original en la que ha cambiando los caracteres iguales a carAnterior por carNuevo. La cadena original no se modifica. También produce el cambio si los parámetros son una secuencia de caracteres.    String   replaceAll(String regex, String str2)   Cambia la todas las apariciones de la de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no cambia.	boolean	match(String regex)	
caracteres iguales a carAnterior por carNuevo. La cadena original no se modifica. También produce el cambio si los parámetros son una secuencia de caracteres.    String   replaceAll(String regex, String str2)   Cambia la todas las apariciones de la de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no cambia.    String   replaceFirst(String regex, String str2)   Cambia la primera aparición de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.    String   substring(int primeraPos, intsegundaPos)   Devuelve true si la cadena que lo invoca comienza con el texto s.    String   substring(int primeraPos, intsegundaPos)   Devuelve el texto que va desde primeraPos a segundaPos. Empieza en 0 y segundaPos no entra.    String   toLowerCase()   Convierte la cadena a minúsculas. La original no se modifica.    String   toUpperCase()   Convierte la cadena a mayúsculas. La original no se modifica.    String   trim()   Elimina los blancos que tenga la cadena tanto al principio delante como al final, sin modificar la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.    String   S			Devuelve una cadena idéntica a la original en la que ha cambiando los
string replace(char carAnterior, char carNuevo)  se modifica. También produce el cambio si los parámetros son una secuencia de caracteres.  Cambia la todas las apariciones de la de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no cambia.  Cambia la primera aparición de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.  Devuelve true si la cadena que lo invoca comienza con el texto s.  String substring(int primeraPos, intsegundaPos)  Cambia la primera aparición de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve el texto que va desde primeraPos a segundaPos. Empieza en 0 y segundaPos no entra.  Char[] toCharArray()  Devuelve un array de caracteres a partir de la cadena dada.  String toLowerCase()  Convierte la cadena a minúsculas. La original no se modifica.  String toUpperCase()  Convierte la cadena a mayúsculas. La original no se modifica.  Elimina los blancos que tenga la cadena tanto al principio delante como al final, sin modificar la cadena que lo invoca.  String split(String regex)  Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.  Devuelve la cadena con el contento de la expresión regular.	String	replace(char carAnterior, char carNuevo)	
String replaceAll(String regex, String str2)  String replaceFirst(String regex, String str2)  String substring(int primeraPos, intsegundaPos)  String toCharArray()  String toLowerCase()  String tring(  String string(  String toLowerCase()  String toUpperCase()  String string(  String toUpperCase()  String tring(  String(  String tring(  Str			
String   replaceAnt(String regex, String str2)   (puede ser una cadena) por la cadena str2. La original no cambia.			
String   replaceFirst(String regex, String str2)   Cambia la primera aparición de la expresión regular regex (puede ser una cadena) por la cadena str2. La original no se modifica.    boolean   startsWith(String s)   Devuelve true si la cadena que lo invoca comienza con el texto s.	64 *	monto o All(Stains and Stains at 2)	Cambia la todas las apariciones de la de la expresión regular <i>regex</i>
toCharArray()  String  toLowerCase()  String  toUpperCase()  String  trim()  String  split(String regex)  split(String regex)  string  to Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve true si la cadena que lo invoca comienza con el texto s.  Devuelve el texto que va desde primeraPos a segundaPos. Empieza en 0 y segundaPos no entra.  Devuelve un array de caracteres a partir de la cadena dada.  Convierte la cadena a minúsculas. La original no se modifica.  Convierte la cadena a mayúsculas. La original no se modifica.  Elimina los blancos que tenga la cadena tanto al principio delante como al final, sin modificar la cadena que lo invoca.  Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.  Devuelve la cadena con el contenido de alemento. Si elemento es	String	replaceAn(String regex, String str2)	(puede ser una cadena) por la cadena str2. La original no cambia.
boolean startsWith(String s)  String substring(int primeraPos, intsegundaPos)  Char[] toCharArray()  String toLowerCase()  String toUpperCase()  String trim()  String substring(int primeraPos, intsegundaPos)  String toUpperCase()  String toUpperCase()  String toUpperCase()  String trim()  S	CAnin a	wonload First (String room, String str2)	Cambia la primera aparición de la expresión regular regex (puede ser
Stringsubstring(int primeraPos, intsegundaPos)Devuelve el texto que va desde primeraPos a segundaPos. Empieza en 0 y segundaPos no entra.char[]toCharArray()Devuelve un array de caracteres a partir de la cadena dada.StringtoLowerCase()Convierte la cadena a minúsculas. La original no se modifica.StringtoUpperCase()Convierte la cadena a mayúsculas. La original no se modifica.Stringtrim()Elimina los blancos que tenga la cadena tanto al principio delante como al final, sin modificar la cadena que lo invoca.String[]split(String regex)Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.	String	replacer irst(String regex, String str2)	una cadena) por la cadena str2. La original no se modifica.
String       substring(int primeraPos, intsegundaPos)       en 0 y segundaPos no entra.         char[]       toCharArray()       Devuelve un array de caracteres a partir de la cadena dada.         String       toLowerCase()       Convierte la cadena a minúsculas. La original no se modifica.         String       trim()       Elimina los blancos que tenga la cadena tanto al principio delante como al final, sin modificar la cadena que lo invoca.         String[]       split(String regex)       Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.	boolean	startsWith(String s)	Devuelve <b>true</b> si la cadena que lo invoca comienza con el texto <i>s</i> .
char[] toCharArray()  Devuelve un array de caracteres a partir de la cadena dada.  String  toLowerCase()  Convierte la cadena a minúsculas. La original no se modifica.  String  trim()  String  trim()  String  String  String  String  String  Convierte la cadena a mayúsculas. La original no se modifica.  Elimina los blancos que tenga la cadena tanto al principio delante como al final, sin modificar la cadena que lo invoca.  Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.	String	substring(int primeraPos intsegundaPos)	
String       toLowerCase()       Convierte la cadena a minúsculas. La original no se modifica.         String       toUpperCase()       Convierte la cadena a mayúsculas. La original no se modifica.         String       trim()       Elimina los blancos que tenga la cadena tanto al principio delante como al final, sin modificar la cadena que lo invoca.         String[]       split(String regex)       Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.	our mg		· -
String toUpperCase()  Convierte la cadena a mayúsculas. La original no se modifica.  Elimina los blancos que tenga la cadena tanto al principio delante como al final, sin modificar la cadena que lo invoca.  String[] split(String regex)  String  Devuelve la cadena con el contenido de elemento. Si elemento es	char[]	toCharArray()	Devuelve un array de caracteres a partir de la cadena dada.
String  trim()  Elimina los blancos que tenga la cadena tanto al principio delante como al final, sin modificar la cadena que lo invoca.  String[]  split(String regex)  String  Devuelve la cadena con el contenido de elemento. Si elemento es	String	toLowerCase()	Convierte la cadena a minúsculas. La original no se modifica.
String   como al final, sin modificar la cadena que lo invoca.  String   split(String regex)   Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.  String   Devuelve la cadena con el contenido de elemento. Si elemento es	String	toUpperCase()	Convierte la cadena a mayúsculas. La original no se modifica.
String   como al final, sin modificar la cadena que lo invoca.  String   split(String regex)   Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.  String   Devuelve la cadena con el contenido de elemento. Si elemento es	G. ·	. • ^	Elimina los blancos que tenga la cadena tanto al principio delante
String   Separa en trozos la cadena que lo invoca. Los trozos se generan cada vez que se encuentra un elemento de la expresión regular.  String   Devuelve la cadena con el contenido de elemento. Si elemento es	String	trim()	
spiri(string regex)  vez que se encuentra un elemento de la expresión regular.  String  Devuelve la cadena con el contenido de elemento. Si elemento es	~		
String Devuelve la cadena con el contenido de elemento. Si elemento es	String[]	<pre>split(String regex)</pre>	· · · · · · · · · · · · · · · · · · ·
String volve (Military algorithm)	String	0.1	
(static)   String.valueOf(tipo elemento)   booleano, devolvería una cadena con el valor true o false.		String.valueOf(tipo elemento)	