


# Capítulo V

## Ejercicio 02 : ft\_str\_is\_alpha

	Ejercicio: 02
ft_str_is_alpha	
Directorio de entrega: <i>ex02/</i>	
Archivos a entregar: <b>ft_str_is_alpha.c</b>	
Funciones autorizadas: Ninguna	


- Crea una función que devuelva 1 si el string usado como parámetro contiene únicamente caracteres alfabéticos y devuelva 0 si contiene otro tipo de caracteres.
- El prototipo de la función deberá ser el siguiente:

```
int      ft_str_is_alpha(char *str);
```

- Tendrá que devolver 1 si **str** es un string vacío.

# Capítulo VI

## Ejercicio 03 : ft\_str\_is\_numeric

	Ejercicio: 03
	ft_str_is_numeric
	Directorio de entrega: <i>ex03/</i>
	Archivos a entregar: <b>ft_str_is_numeric.c</b>
	Funciones autorizadas: Ninguna


- Crea una función que devuelva 1 si el string usado como parámetro contiene únicamente dígitos y devuelva 0 si contiene otro tipo de caracteres.
- El prototipo de la función deberá ser el siguiente:

```
int ft_str_is_numeric(char *str);
```

- Tendrá que devolver 1 si **str** es un string vacío.

# Capítulo VII

## Ejercicio 04 : ft\_str\_is\_lowercase

	Ejercicio: 04
	ft_str_is_lowercase
	Directorio de entrega: <i>ex04/</i>
	Archivos a entregar: <b>ft_str_is_lowercase.c</b>
	Funciones autorizadas: Ninguna


- Crea una función que devuelva 1 si el string usado como parámetro contiene únicamente caracteres alfabéticos en minúsculas y devuelva 0 si contiene otro tipo de caracteres.
- El prototipo de la función deberá ser el siguiente:

```
int ft_str_is_lowercase(char *str);
```

- Tendrá que devolver 1 si **str** es un string vacío.

# Capítulo VIII

## Ejercicio 05 : ft\_str\_is\_uppercase

	Ejercicio: 05
ft_str_is_uppercase	
Directorio de entrega: <i>ex05/</i>	
Archivos a entregar: <b>ft_str_is_uppercase.c</b>	
Funciones autorizadas: Ninguna	


- Crea una función que devuelva 1 si el string usado como parámetro contiene únicamente caracteres alfabéticos en mayúsculas y devuelva 0 si contiene otro tipo de caracteres.
- El prototipo de la función deberá ser el siguiente:

```
int ft_str_is_uppercase(char *str);
```

- Tendrá que devolver 1 si **str** es un string vacío.

# Capítulo IX

## Ejercicio 06 : ft\_str\_is\_printable

	Ejercicio: 06
ft_str_is_printable	
Directorio de entrega: <i>ex06/</i>	
Archivos a entregar: <b>ft_str_is_printable.c</b>	
Funciones autorizadas: Ninguna	


- Crea una función que devuelva 1 si el string usado como parámetro contiene únicamente caracteres imprimibles y devuelva 0 si contiene otro tipo de caracteres.
- El prototipo de la función deberá ser el siguiente:

```
int ft_str_is_printable(char *str);
```

- Tendrá que devolver 1 si **str** es un string vacío.

# Capítulo X

## Ejercicio 07 : ft\_strupcase

	Ejercicio: 07
	ft_strupcase
	Directorio de entrega: <i>ex07/</i>
	Archivos a entregar: <b>ft_strupcase.c</b>
	Funciones autorizadas: Ninguna


- Crea una función que ponga cada letra en mayúscula.
- El prototipo de la función deberá ser el siguiente:

```
char *ft_strupcase(char *str);
```

- Deberá devolver **str**.

# Capítulo XI

## Ejercicio 08 : ft\_strlowcase

	Ejercicio: 08
ft_strlowcase	
Directorio de entrega: <i>ex08/</i>	
Archivos a entregar: <b>ft_strlowcase.c</b>	
Funciones autorizadas: Ninguna	


- Crea una función que ponga cada letra en minúscula.
- El prototipo de la función deberá ser el siguiente:

```
char      *ft_strlowcase(char *str);
```

- Deberá devolver **str**.

# Capítulo XII

## Ejercicio 09 : ft\_strcapitalize

	Ejercicio: 09
ft_strcapitalize	
Directorio de entrega: <i>ex09/</i>	
Archivos a entregar: <b>ft_strcapitalize.c</b>	
Funciones autorizadas: Ninguna	

- Creaa una función que ponga en mayúscula la primera letra de cada palabra y el resto de la palabra en minúsculas.
- Una palabra es una secuencia (string) de caracteres alfanuméricos.
- El prototipo de la función deberá ser el siguiente:

```
char      *ft_strcapitalize(char *str);
```

- Deberá devolver **str**.
- Por ejemplo:

```
salut, comment tu vas ? 42mots quarante-deux; cinquante+et+un
```

- Se convierte en:

```
Salut, Comment Tu Vas ? 42mots Quarante-Deux; Cinquante+Et+Un
```