

A large, light gray, 3D-style letter 'R' logo that serves as a background for the text. The 'R' has a thick, rounded top and a vertical stem with a diagonal base.

BIE5782

Unidade 2:

FUNÇÕES NUMÉRICAS

Operadores + , - , * , / , ^

Funções Aritméticas

```
> 4 + 9
```

```
[1] 13
```

```
> 4 - 5
```

```
[1] -1
```

```
> 4 * 5
```

```
[1] 20
```

```
> 4 / 5
```

```
[1] 0.8
```

```
> 4^5
```

```
[1] 1024
```

Precedência de Operações e Parênteses

> 2*4^3 - 1

[1] 127

> 2*4^(3 - 1)

[1] 32

> (2*4)^3 - 1

[1] 511

> (2*4)^(3 - 1)

[1] 64

sqrt() , abs()

Algumas Funções Matemáticas

```
> sqrt(9)      # Raiz Quadrada  
[1] 3
```

```
> abs(-1)     # Módulo ou valor absoluto  
[1] 1
```

log(), exp()

Logarítmo e exponenciação

```
> log( 10 )      # Logaritmo natural  
[1] 2.302585
```

```
> log( 10, base = 10) # Log base 10  
[1] 1
```

```
> log10(10)     # Também log de base 10  
[1] 1
```

```
> log( 10, base = 3.4076) # base 3.4076  
[1] 1.878116
```

```
> exp( 1 )      # Exponencial  
[1] 2.718282
```

π , $\sin()$, $\cos()$, $\tan()$...

Constante π e Funções Trigonométricas

```
> sin(0.5*pi)      # Seno  
[1] 1
```

```
> cos(2*pi)       # Coseno  
[1] 1
```

```
> asin(1)         # Arco seno (radianos)  
[1] 1.570796
```

```
> asin(1) / pi * 180  
[1] 90
```

factorial(), **gamma()**, **choose()**,

... O R é uma super-calculadora!

```
> 1 - (1 + 10^(-15))
```

```
[1] -1.110223e-15
```

```
> factorial(100) # Fatorial de 100
```

```
[1] 9.332622e+157
```

```
> gamma(101) #Função gama
```

```
[1] 9.332622e+157
```

```
> choose(1.93e+8, 2) #N casais no Brasil
```

```
[1] 1.86245e+16
```

A Terra tem $\sim 10^{50}$ átomos e o universo $\sim 10^{24}$ estrelas

ceiling() , floor() , round()

Arredondamento

```
> ceiling( 4.3478 )  
[1] 5  
> floor( 4.3478 )  
[1] 4  
> round( 4.3478 )  
[1] 4  
> round( 4.3478 , digits=3)  
[1] 4.348  
> round( 4.3478 , digits=2)  
[1] 4.35
```

CUIDADO: são funções, e não formatação de decimais a exibir

Atribuição e Variáveis Pré-definidas

```
> a <- 3.6
> b <- sqrt( 35 )
> c <- -2.1
> a
[1] 3.6
> b
[1] 5.91608
> c
[1] -2.1
>
> a * b / c
[1] -10.14185
> a - b * c / d
Error: object "d" not found
```

Não foi definida!



NA , NaN , Inf , -Inf

Valores Infinitos, Indefinidos e Inexistentes

```
> -5/0
[1] -Inf
> 50000000000000000000000000000000/Inf
[1] 0
> sqrt( - 1 )
[1] NaN
Warning message:
NaNs produced in: sqrt(-1)
> 2 * NA
[1] NA
> 2 * NaN
[1] NaN
```

Senta que lá vem história!



VETORES E OPERAÇÕES VETORIAIS

c()

Criação de Vetores

```
> a = c(3.4, pi, exp(-1))  
> a  
[1] 3.40000000 3.1415927 0.3678794
```

- Um objeto da classe `vector` no R é um conjunto de elementos de uma mesma classe (números, caracteres, etc).
- Um objeto vetor no R não corresponde aos vetores de álgebra matricial (para isso há a classe `matrix`).
- A função `c` combina elementos em um vetor.

▪
▪

Criação de Sequências

```
> b = 1:8
```

```
> b
```

```
[1] 1 2 3 4 5 6 7 8
```

```
> 2.5:10
```

```
[1] 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5
```

seq()

Criação de Sequências

```
> seq(from=1, to=4)      # mesmo que 1:4  
[1] 1 2 3 4
```

```
> seq(from=1, to=4, by=0.5)  
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0
```

```
> seq(from=1, to=4, length=6)  
[1] 1.0 1.6 2.2 2.8 3.4 4.0
```

rep()

Repetições

```
> rep(5, times=3)  
[1] 5 5 5
```

```
> rep(1:5, 3)  
[1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

```
> rep(1:5, each=3)  
[1] 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5
```

Operações com um Vetor

```
> a = seq(0, 8, 2)
```

```
> a
```

```
[1] 0 2 4 6 8
```

```
> 2*a
```

```
[1] 0 4 8 12 16
```

```
> sqrt(a)
```

```
[1] 0.0000000 1.414214 2.0000000 2.449490
```

```
[5] 2.828427
```

Todas as operações aplicadas a um vetor são aplicadas a cada um de seus elementos

Operações entre Vetores

```
> a = seq(0, 8, 2)
```

```
> a
```

```
[1] 0 2 4 6 8
```

```
> b = c(1, 15, 18, 3, 6)
```

```
> a+b
```

```
[1]  1 17 22  9 14
```

```
> a^(1/b)
```

```
[1] 0.0000000 1.047294 1.080060 1.817121
```

```
[5] 1.414214
```

Operações entre vetores: pareando os elementos

A REGRA DA CICLAGEM

```
> b
```

```
[1] 0 0 0 0 0 1 1 1 1 1
```

```
> c
```

```
[1] 1 2 3
```

```
> c*b
```

```
[1] 0 0 0 0 0 3 1 2 3 1
```

Warning message:

**In c * b : longer object length is not
a multiple of shorter object length**

Os elementos do vetor mais curto são repetidos sequencialmente até que a operação seja aplicada a todos os elementos do vetor mais longo.

length()

Comprimento de Vetores

```
> a
```

```
[1] 1 2
```

```
> b
```

```
[1] 0 0 0 0 0 1 1 1 1 1
```

```
> a*b
```

```
[1] 0 0 0 0 0 2 1 2 1 2
```

```
> length(b)/length(a)
```

```
[1] 5
```

mean, var, sd, min, max, range, sum ...

Funções Estatísticas

```
> b = c(1, 15, 18, 3, 6)
```

```
> mean(b)
```

```
[1] 8.6
```

```
> var(b)
```

```
[1] 56.3
```

```
> min(b)
```

```
[1] 1
```

```
> max(b)
```

```
[1] 18
```

```
> sum(b)
```

```
[1] 43
```

Algumas funções, como estas, operam sobre **todo** o vetor, e não elemento a elemento.

`cumsum()`, `sort()`, `diff()`

Também Operam sobre Todo o Vetor

```
> b
[1]  1 15 18  3  6
> cumsum(b)
[1]  1 16 34 37 43
> sort(b)
[1]  1  3  6 15 18
> sort(b, decreasing=T)
[1] 18 15  6  3  1
> diff(b)
[1] 14  3 -15  3
```

Algumas funções, como estas, operam sobre **todo** o vetor, e não elemento a elemento.

tapply()

Funções por níveis dos Fatores

```
> sexo
```

```
[1] F F F F F F F F F M M M M M M M M
```

```
Levels: F M
```

```
> peso <-
```

```
c(66, 70, 67, 79, 82, 54, 61, 59, 63, 85, 78, 112, 92, 77, 82,  
+ 85, 80, 91)
```

```
> tapply(X=peso, INDEX=sexo, FUN=mean)
```

```
      F      M  
66.77778 86.88889
```

tapply()

Funções por níveis dos Fatores

```
> sexo
```

```
[1] F F F F F F F F F M M M M M M M M
```

```
Levels: F M
```

```
> peso <-
```

```
c(66, 70, 67, 79, 82, 54, 61, 59, 63, 85, 78, 112, 92, 77, 82,  
+ 85, 80, 91)
```

```
> tapply(X=peso, INDEX=sexo, FUN=range)
```

```
$F
```

```
[1] 54 82
```

```
$M
```

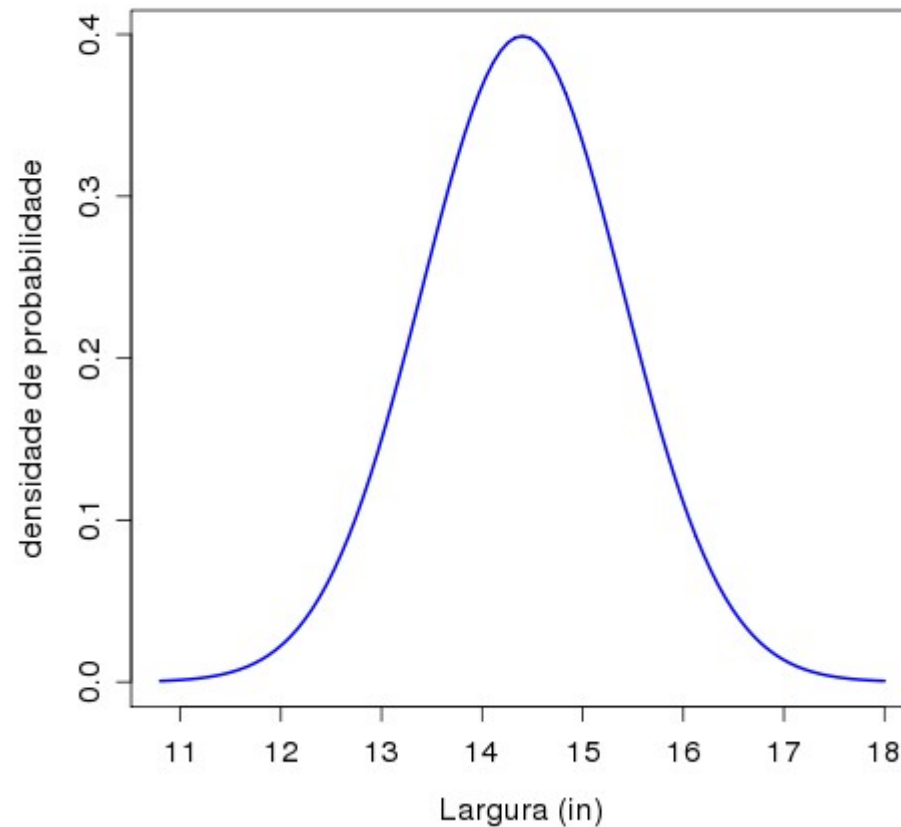
```
[1] 77 112
```

Senta que lá vem MAIS história!



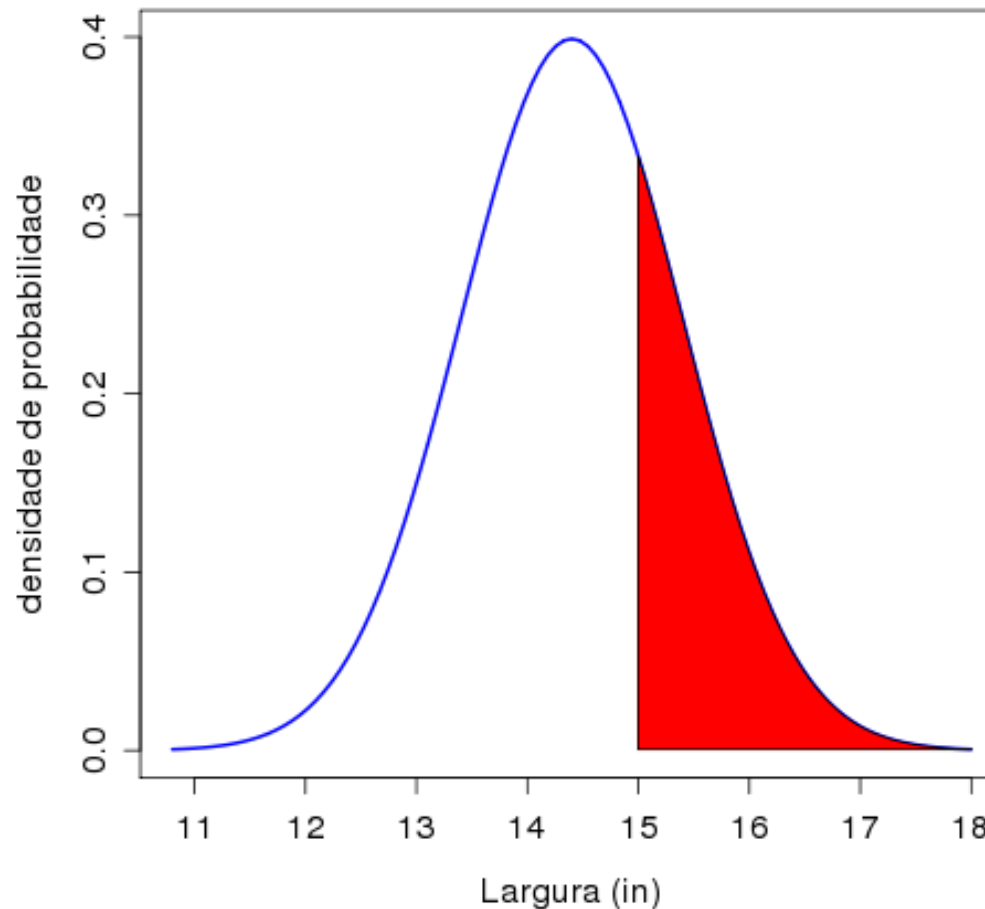
DISTRIBUIÇÕES DE PROBABILIDADES NO R

Um Exemplo Verídico: Quadris lanques



Segundo as Cias. Aéreas dos EUA, a largura dos quadris dos homens norte-americanos segue uma curva normal, com média de 14,4 polegadas, e desvio-padrão de 1,0 polegada

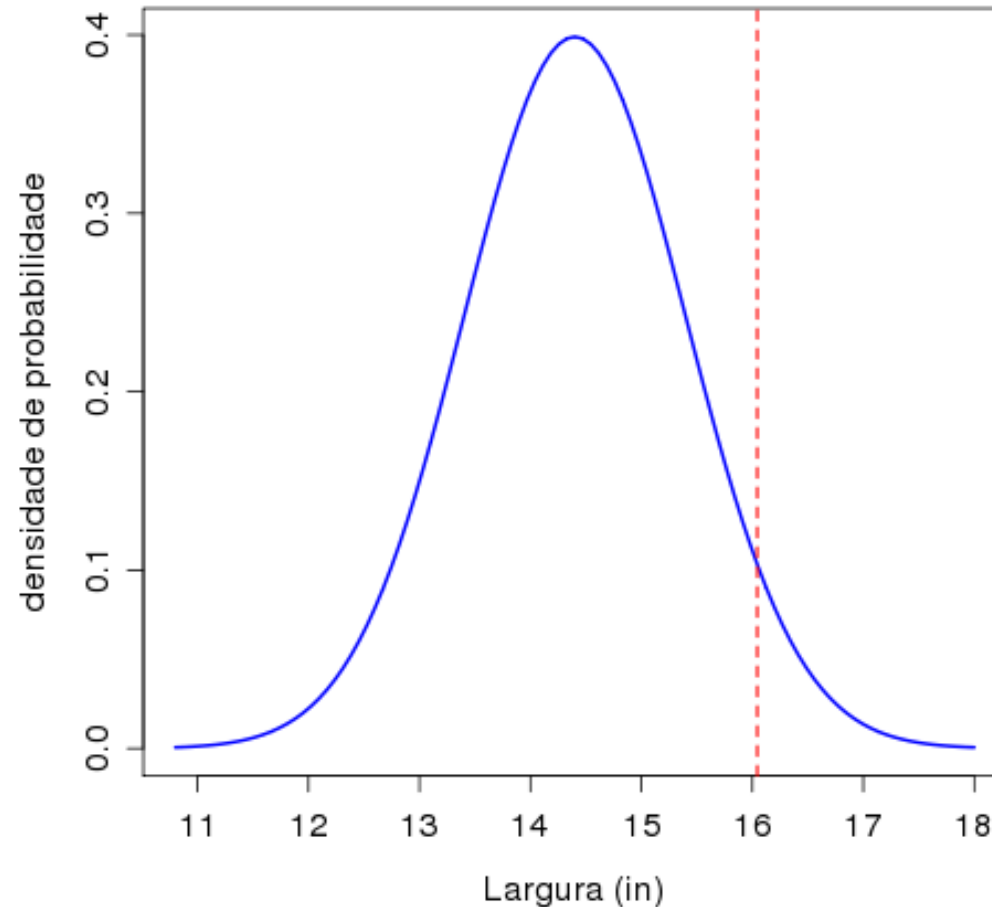
Qual o Percentual da População não Cabe em um Assento de 15 polegadas?



Note este argumento!

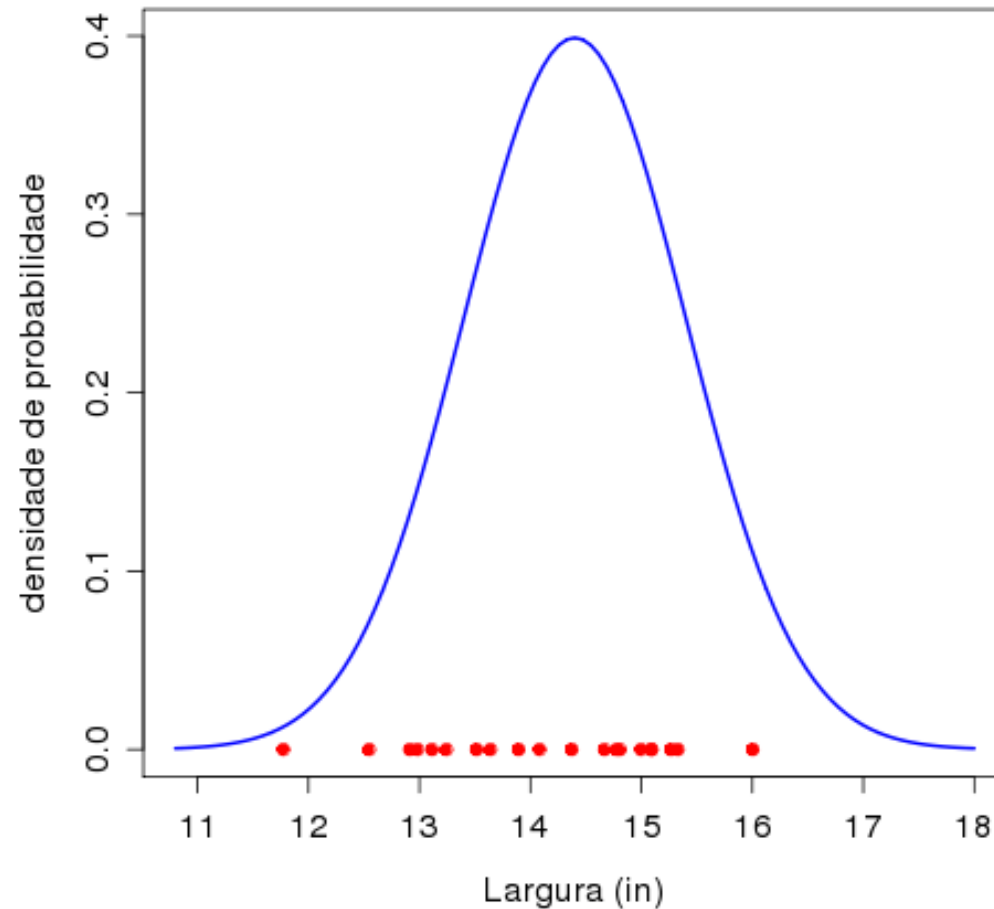
```
> pnorm(q=15, mean=14.4, sd=1, lower.tail=F)
[1] 0.2742531
```

Qual Largura Garante que 95% da População Caberá?



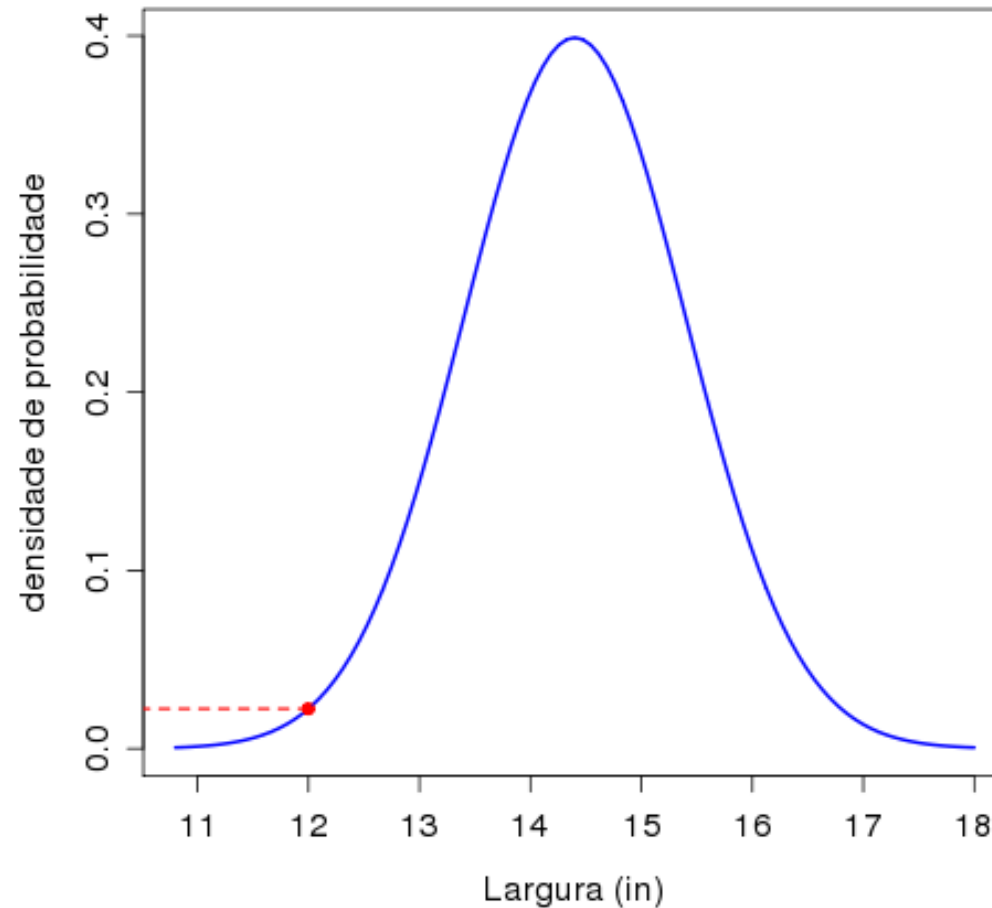
```
> qnorm(p=0.95, mean=14.4, sd=1)  
[1] 16.04485
```

Simule uma Amostra de 20 Passageiros



> **rnorm(n=20, mean=14.4, sd=1)**

Qual a Densidade Probabilística do valor de Largura de Quadril = 12 in ?



```
> dnorm(x=12, mean=14.4, sd=1)  
[1] 0.02239453
```

Distribuições: Família **d** , **p** , **q** , **r**

Normal

`dnorm()`, `pnorm()`, `qnorm()`, `rnorm()`

t

`dt()`, `pt()`, `qt()`, `rt()`

Qui-quadrado

`dchisq()`, `pchisq()`, `qchisq()`, `rchisq()`

Poisson

`dpois()`, `ppois()`, `qpois()`, `rpois()`

...

Algumas Distribuições no R

Distribuição	Nome no R	Parâmetros ³⁾
beta	beta	shape1, shape2, ncp
binomial	binom	size, prob
Cauchy	cauchy	location, scale
qui-quadrado	chisq	df, ncp
exponential	exp	rate
F	f	df1, df2, ncp
gamma	gamma	shape, scale
geométrica	geom	prob
hipergeométrica	hyper	m, n, k
log-normal	lnorm	meanlog, sdlog
logística	logis	location, scale
binomial negativa	nbinom	size, prob
normal	norm	mean, sd
Poisson	pois	lambda
t de Student	t	df, ncp
uniforme	unif	min, max
Weibull	weibull	shape, scale
Wilcoxon	wilcox	m, n

Fim da Unidade 2

Agora:

Plantão tutoriais e exercícios – Lab. Bioquímica I

Postar no wiki até quarta:

Listas 1 e 2 dos exercícios

http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:01_curso_atual:exercicios1

http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:01_curso_atual:exercicios2