



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico Número 1

Especificación

Algoritmos y Estructuras de Datos I

Grupo: 4

Integrante	LU	Correo electrónico
Aun Castells, María Virginia	366/13	vauncastells@hotmail.com
Motta, Leandro	85/14	leamotta@msn.com
Zdanovitch, Nikita	520/14	3hb.tch@gmail.com
de Monasterio, Francisco	764/13	franciscodemonasterio@outlook.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

1. Observaciones

1. un item
2. otro item

2. Resolución

Ejercicio 1. Blur:

```
problema blur(imagen : [[(Z, Z, Z)]], k : Z) = res : [[(Z, Z, Z)]]{
  requiere kEsPositivo: k > 0;
  requiere entradaEsRectangular: EsRectangular(imagen);
  requiere todosLosPixelsSonValidos: (∀f ← imagen)(∀px ← f)();
  asegura dimensionesDeSalidaIguales: Alto(res) == Alto(imagen) ∧ Ancho(res) ==
    Ancho(imagen);
  asegura aseguraTodo: (∀y ← [0 :| Alto(imagen) |])(∀x ← [0 :| Ancho(imagen) |
    ]) if esKCompleto(KVecinos(imagen, y, x, k), k) then esPromedio(res, imagen, y, x, k) else esNegativo(res, y
  }
```

Ejercicio 2. Dividir:

```
problema dividir(imagen : [[(Z, Z, Z)]], m, n : Z) = res : [[(Z, Z, Z)]]{
  requiere nYmEsPositivo: n > 0 ∧ m > 0;
  requiere entradaTieneSuperficie: Alto(imagen) > 0 ∧ Ancho(imagen) > 0;
  requiere todosLosPixelsSonValidos: (∀f ← imagen)(∀px ← f);
  requiere divideEnFilasIguales: Alto(imagen) mód m == 0;
  requiere divideEnColumnasIguales: Ancho(imagen) mód n == 0;
  asegura : mismo(res, SepararHorizontal(SepararVertical(imagen, n), m));
}
```

2.1. Auxiliares

- aux Alto(img : [[(Z, Z, Z)]] : Z = |img|;
- aux Ancho(img : [[(Z, Z, Z)]] : Z = if |img| == 0 then 0 else |cab(img)|;
- aux esIndiceValido(y, x : Z, img : [[(Z, Z, Z)]] : Bool = 0 ≤ y < Alto(img) ∧ 0 ≤ x < Ancho(img);
- aux esRectangular(img : [[(Z, Z, Z)]] : Bool = (∀a ← img) |a| == |cab(img)|;
- aux KIndices(y, x, k : Z : [(Z, Z)] = [(i, j) | i ← (y - k : y + k), j ← (x - k : x + k)];
- aux KVecinos(img : [[(Z, Z, Z)]], y, x, k : Z : [(Z, Z, Z)] = [Pixel(img, prm(c), sgd(c)) | c ← KIndices(y, x, k), esIndiceValido(c)];
- aux Pixel(img : [[(Z, Z, Z)]], y, x : Z : (Z, Z, Z) = if esIndiceValido(y, x, img) then img[y][x] else (0, 0, 0);
- aux EsNegro(r : [[(Z, Z, Z)]], y, x : Z : Bool = prm(Pixel(r, y, x)) == 0 ∧ sgd(Pixel(r, y, x)) == 0 ∧ trc(Pixel(r, y, x)) == 0;
- aux EsPromedio(r, img : [[(Z, Z, Z)]], y, x, k : Z : Bool = prm(Pixel(r, y, x)) == (∑ prm(p) | p ← KVecinos(img, y, x, k)) / KVecinos(img, y, x, k) ∧ sgd(Pixel(r, y, x)) == (∑ sgd(p) | p ← KVecinos(img, y, x, k)) / KVecinos(img, y, x, k) ∧ trc(Pixel(r, y, x)) == (∑ trc(p) | p ← KVecinos(img, y, x, k)) / KVecinos(img, y, x, k);
- aux EsKCompleto(kv : [(Z, Z, Z)], k : Z : Bool = |kv| == (k + k - 1)²;

- **aux** *VektorKVecinos*(*img* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$, *i, j, k* : \mathbb{Z}) : $(\mathbb{Z}, \mathbb{Z}, \mathbb{Z}) = [img[a][b] \mid a \leftarrow [i - k + 1 : i + k - 1], b \leftarrow [j - k + 1 : j + k - 1], a \geq 0 \wedge b \geq 0 \wedge a < Alto(img) \wedge b < Ancho(img)]$;
- **aux** *MedianaONegro*(*img* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$, *i, j, k* : \mathbb{Z}) : $(\mathbb{Z}, \mathbb{Z}, \mathbb{Z}) = if esKCompleto(vectorKVecinos(img, k, i, j)) then$
- **aux** *mediana*(*vector* : $(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})$) : $(\mathbb{Z}, \mathbb{Z}, \mathbb{Z}) = (valorMediana([prm(a) \mid a \leftarrow vector], valorMediana([sgd(a) \mid a \leftarrow vector], valorMediana([trc(a) \mid a \leftarrow vector]))$;
- **aux** *valorMediana*(*xs* : $[\mathbb{Z}]$) : $\mathbb{Z} = enOrden(xs)[|xs| \div 2]$;
- **aux** *enOrden*(*xs* : $[\mathbb{Z}]$) : $[\mathbb{Z}] = [x \mid i \leftarrow [0 : |xs|), x \leftarrow xs, cuentaMenores(xs, x) == i]$;
- **aux** *cuentaMenores*(*xs* : $[\mathbb{Z}]$, *x* : \mathbb{Z}) : $\mathbb{Z} = |[1 \mid y \in xs, y < x]|$;
- **aux** *SepararVertical*(*img* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$, *columnas* : \mathbb{Z}) : $[[[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]]] = [verticalizarImagen(img, columnas)[i * Alto(img)..(i + 1) * Alto(img)] \mid i \leftarrow [0..columnas)]$;
- **aux** *verticalizarImagen*(*img* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$, *columnas* : \mathbb{Z}) : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})] = [img[i][Ancho(img) * k \div columnas..Ancho(img) * (k + 1 \div columnas)] \mid k \leftarrow [0..columnas), i \leftarrow [0..Alto(img)]]$;
- **aux** *SepararHorizontal*(*listaimg* : $[[[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]]]$, *filas* : \mathbb{Z}) : $[[[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]]] = [listaimg[i][|cab(listaimg)| * k \div filas..|cab(listaimg)| * (k + 1 \div filas) \mid k \leftarrow [0..filas), i \leftarrow [0..|listaimg|)] \mid a \leftarrow [i - k + 1 : i + k - 1], b \leftarrow [j - k + 1 : j + k - 1], a \geq 0 \wedge b \geq 0 \wedge a < Alto(img) \wedge b < Ancho(img)]$;
- **aux** *cuenta*(*x* : *T*, *a* : $[T]$) : $\mathbb{Z} = long([y \mid y \leftarrow a, y == x])$;
- **aux** *mismos*(*a, b* : $[T]$) : $Bool = (|a| == |b| \wedge (\forall c \leftarrow a) cuenta(c, a) == cuenta(c, b))$;