



**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 ítem
2. otro ítem

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 dimensionesDeSalidalguales: 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, x, k)
}
```

Ejercicio 2. Acuarela:

```
problema acuarela(imagen : [[(Z, Z, Z)]], k : Z) = res : [[(Z, Z, Z)]]{
  requiere kEsPositivo: k > 0;
  requiere entradaEsRectangular: EsRectangular(imagen);
  asegura dimensionesDeSalidalguales: Alto(res) == Alto(imagen) ∧ Ancho(res) ==
    Ancho(imagen);
  asegura efecto: (∀i ← [0 : |Alto(res)|])(∀j ← [0 : |Ancho(res)|]) res[i][j] ==
    MedianaONegro(i, j, img, k);
}
```

Ejercicio 3. 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));
}
```

Ejercicio 4. Pegar:

```
problema pegar(imagen, imagen2 : [[(Z, Z, Z)]], pixel : (Z, Z, Z) = {
  modifica imagen;
  requiere hayRectanguloColorPixelEnImagen: hayRectangulo(pre(imagen), pixel);
  requiere entradaEsRectangular: EsRectangular(pre(imagen));
  requiere entrada2EsRectangular: EsRectangular(imagen2);
  requiere pixelesFormanRectangulo: EsRectangular(rectaPixel(pre(imagen), pixel));
  requiere entradaTieneSuperficie: Alto(pre(imagen)) > 0 ∧ Ancho(pre(imagen)) > 0;
  requiere entrada2TieneSuperficie: Alto(imagen2) > 0 ∧ Ancho(imagen2) > 0;
  asegura imagen2ContenidaEnRect: if Ancho(imagen2) ≤
    Ancho(rectaPixel(pre(imagen), pixel)) ∧ Alto(imagen2) ≤
    Alto(rectaPixel(pre(imagen), pixel)) then imagen ==
    pegoImagenes(pre(imagen), pegoImagenes(rectaPixel(pre(imagen), pixel), imagen2, 0, 0), , ) else pre(imagen);
  asegura imagenTieneMismoTamaño: Alto(pre(imagen)) == Alto(imagen) ∧
    Ancho(pre(imagen)) == Ancho(imagen);
}
```

}

2.1. Auxiliares

- **aux** *Alto*(*img* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$) : $\mathbb{Z} = |img|$;
- **aux** *Ancho*(*img* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$) : $\mathbb{Z} = if \ |img| == 0 \ then \ 0 \ else \ |cab(img)|$;
- **aux** *esIndiceValido*(*y, x* : \mathbb{Z} , *img* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$) : **Bool** = $0 \leq y < Alto(img) \wedge 0 \leq x < Ancho(img)$;
- **aux** *esRectangular*(*img* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$) : **Bool** = $(\forall a \leftarrow img) \ |a| == |cab(img)|$;
- **aux** *KIndices*(*y, x, k* : \mathbb{Z} : $[(\mathbb{Z}, \mathbb{Z})] = [(i, j) \mid i \leftarrow (y - k : y + k), j \leftarrow (y - k : y + k)]$);
- **aux** *KVecinos*(*img* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$, *y, x, k* : \mathbb{Z} : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})] = [Pixel(img, prm(c), sgd(c)) \mid c \leftarrow KIndices(y, x, k), esIndiceValido(c)]$;
- **aux** *Pixel*(*img* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$, *y, x* : \mathbb{Z}) : $(\mathbb{Z}, \mathbb{Z}, \mathbb{Z}) = if \ esIndiceValido(y, x, img) \ then \ img[y][x] \ else \ (0, 0, 0)$;
- **aux** *EsNegro*(*r* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$, *y, x* : \mathbb{Z}) : **Bool** = $prm(Pixel(r, y, x)) == 0 \wedge sgd(Pixel(r, y, x)) == 0 \wedge trc(Pixel(r, y, x)) == 0$;
- **aux** *EsPromedio*(*r, img* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$, *y, x, k* : \mathbb{Z}) : **Bool** = $prm(Pixel(r, y, x)) == (\sum prm(p) \mid p \leftarrow KVecinos(img, y, x, k)) \wedge sgd(Pixel(r, y, x)) == (\sum sgd(p) \mid p \leftarrow KVecinos(img, y, x, k)) \wedge trc(Pixel(r, y, x)) == (\sum trc(p) \mid p \leftarrow KVecinos(img, y, x, k))$;
- **aux** *EsKCompleto*(*kv* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$, *k* : \mathbb{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}) : $(\mathbb{Z}, \mathbb{Z}, \mathbb{Z}) = if \ esKCompleto(vectorKVecinos(img, k, i, j)) \ then \ (valorMediana([prm(a) \mid a \leftarrow vector], valorMediana([sgd(a) \mid a \leftarrow vector], valorMediana([trc(a) \mid a \leftarrow vector]))$;
- **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** = $(|a| == |b| \wedge (\forall c \leftarrow a) \ cuenta(c, a) == cuenta(c, b))$;
- **aux** *hayRectangulo*(*img* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$, *pxl* : $(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})$) : **Bool** = $i \leftarrow [0 .. Alto(img)], j \leftarrow [0 .. Ancho(img)], 4DelMismoColor(vectorKVecinos(img, 2, i, j), pxl)$;
- **aux** *4DelMismoColor*(*kv* : $[(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})]$, *pxl* : $(\mathbb{Z}, \mathbb{Z}, \mathbb{Z})$) : **Bool** = $(\forall f \leftarrow kv)$;