

# Datos importantes para los exámenes de cursada y finales

La cursada requiere la realización, entrega y defensa de un trabajo, el mismo lo puede ver [aquí](#).

El código necesario para generar la tabla es el siguiente:

```
<table border=5>
  <tr>
    <td>Pérez, Juan</td>
    <td>Lógica</td>
    <td><font color='green'>4</font></td>
  </tr> format
  <tr>
    <td>Pérez, Juan</td>
    <td>Álgebra</td>
    <td><font color='green'>7</font></td>
  </tr>
  <tr>
    <td>Pérez, Juan</td>
    <td>Análisis</td>
    <td><font color='red'>2</font></td>
  </tr>
</table>
```

Para aprobar la cursada debe aprobar el Parcial (con 4 o más) y el Trabajo.

Vamos a dar la alternativa de promoción, para alcanzar la misma deberá aprobar el Trabajo con Promoción y también el Parcial (con nota igual o mayor a siete).

## A tener en cuenta antes de comenzar a escribir el examen

- Escriba con birome, no importa si tiene que tachar.
- Escriba de un solo lado en cada hoja.
- No empiece un nuevo módulo en una página que está escrita si ve que no lo va a terminar en la misma página, comience en una nueva.

## Preguntas a hacerse antes de entregar el examen

- ¿He asumido alguna condición especial que no está claramente expresada en el enunciado? Si es así ¿incorporé un comentario aclarándolo?
- ¿Hice el diagrama de estructura (DE)?
- ¿Le puse todas las cuplas?
- En las funciones ¿le puse la cupla de retorno?
- ¿Las funciones tienen nombre “de función” ?

- ¿Los “procedimientos” tienen nombre de acción?
- ¿Respeté los nombres del DE y los del código? ¿Son los mismos? ¿Sobran? ¿Faltan?
- He indentado todo el código?
- Analizo cada módulo (procedimiento o función). ¿Hay alguno que realiza más de una tarea conceptualmente? ¿Es conveniente dividirlo?
- Tengo algún módulo que realice gestión y acciones al mismo tiempo? Sería conveniente dividirlo?
- ¿Las cuplas del DE se corresponden con todos los parámetros? ¿Sobran? ¿Faltan?
- ¿Todas las cuplas de salida del DE están pasadas por referencia (utilizando &) en los módulos? ¿Y viceversa?
- ¿Todas las variables/arreglos tienen nombres representativos?
- ¿Todos los módulos tienen nombres representativos?
- ¿Incorporé comentarios en el código?
- ¿Tengo definida alguna variable o parámetro en alguna función que no uso? ¿Y alguna que uso y no definí?
- Dentro de cada código, programa o módulo, ¿definí e inicialicé todas las variables que uso al principio?

## Una alternativa de resolución del Prefinal de la cursada 2012

```
<?php
define('SEP_CAMPOS_ARCH', ';');
define('NOMBRE_ARCH_SUCURSALES', 'sucursales.txt');
define('ARCH_SUCURSALES_POS_CAMPO_PROVINCIA', 1);
define('ARCH_SUCURSALES_POS_CAMPO_ARCHVENTAS', 3);
define('ARCH_VENTAS_POS_CAMPO_FECHA', 1);
define('ARCH_VENTAS_POS_CAMPO_IMPORTE', 2);

function anio($fecha) {
    return intval(substr($fecha, 6));
}

function anio_actual() {
    $hoy = getdate();
    return $hoy['year'];
}

function acumular_ventas_anio($nombre_archivo, $anio, &$cantidad_ventas,
    &$facturacion_total) {
    $cantidad_ventas = 0;
    $facturacion_total = 0;
    $archivo = @fopen($nombre_archivo, 'r');
    if ($archivo) {
        if (!feof($archivo)) { // El archivo puede estar vacío
            $linea = fgets($archivo); // Se saltea la cabecera
        }
        $anio_venta = 0;
        while (!feof($archivo) && $anio_venta <= $anio) {
```

```

        $linea = trim(fgets($archivo));
        if ($linea != '') { // Se saltean posibles líneas en blanco
            $campos = array_map('trim', explode(SEP_CAMPOS_ARCH,
$linea));
            $anio_venta = anio($campos[ARCH_VENTAS_POS_CAMPO_FECHA]);
            if ($anio_venta == $anio) {
                $facturacion_total +=
$campos[ARCH_VENTAS_POS_CAMPO_IMPORTE];
                $cantidad_ventas++;
            }
        }
    }
    fclose($archivo);
}

function mostrar_cabecera_estadisticas_venta_provincia() {
    print 'Provincia | Total | Promedio' . PHP_EOL;
    print '-----' . PHP_EOL;
}

function mostrar_estadisticas_ventas_provincia($provincia, $cantidad_ventas,
$facturacion_total) {
    $promedio = $cantidad_ventas != 0 ? $facturacion_total /
$cantidad_ventas : 0;
    printf("%s | %.2f | %.2f" . PHP_EOL, $provincia, $facturacion_total,
$promedio);
}

function mostrar_estadisticas_ventas_por_provincia() {
    $arch_sucursales = @fopen(NOMBRE_ARCH_SUCURSALES, 'r');
    if ($arch_sucursales) {
        mostrar_cabecera_estadisticas_venta_provincia();
        if (!feof($arch_sucursales)) { // El archivo puede estar vacío
            $linea_sucursales = fgets($arch_sucursales); // Se saltea la
cabecera
            $provincia_actual = NULL;
            $anio_pasado = anio_actual() - 1;
            while (!feof($arch_sucursales)) {
                $linea_sucursales = trim(fgets($arch_sucursales));
                if ($linea_sucursales != '') { // Se saltean posibles líneas
en blanco
                    $campos_sucursales = array_map('trim',
explode(SEP_CAMPOS_ARCH, $linea_sucursales));
                    $provincia =
$campos_sucursales[ARCH_SUCURSALES_POS_CAMPO_PROVINCIA];

                    /*
                     * El archivo está ordenado por provincia, por lo que se
pueden
                     * ir calculando las estadísticas de cada provincia a

```

```
medida
        * que se lee.
        */

        if ($provincia != $provincia_actual) {
            /*
            de leer
            reinician.
            */

            // En la 1er pasada por el while, $provincia_actual
            es NULL
            // y no hay datos para mostrar aún
            if (!is_null($provincia_actual)) {
mostrar_estadisticas_ventas_provincia($provincia_actual, $cantidad_ventas,
$facturacion_total);
            }
            $cantidad_ventas = 0;
            $facturacion_total = 0;
            $provincia_actual = $provincia;
        }

acumular_ventas_anio($campos_sucursales[ARCH_SUCURSALES_POS_CAMPO_ARCHVENTAS
],
                    $anio_pasado, $cantidad_ventas_sucursal,
$facturacion_total_sucursal);
        $cantidad_ventas += $cantidad_ventas_sucursal;
        $facturacion_total += $facturacion_total_sucursal;
    }
}
if (!is_null($provincia_actual)) { // Va a ser NULL sólo si el
archivo no tiene datos
    // Cuando se termina el archivo hay que mostrar la última
provincia que se estaba leyendo
    mostrar_estadisticas_ventas_provincia($provincia_actual,
$cantidad_ventas, $facturacion_total);
}
}
fclose($arch_sucursales);
}
}

mostrar_estadisticas_ventas_por_provincia();
?>
```

## Una alternativa de resolución del Recuperatorio de la cursada 2012

### Enunciado

#### Enunciado del Recuperatorio

```
<?php
define('ARCHIVO_DATOS' , 'datos.txt');
define('ARCHIVO_SALIDA', 'salida.txt');
define('POS_ESTACION', 0);
define('POS_FECHA', 1);
define('POS_TIPO' , 2);
define('POS_VALOR', 3);
define('POS_MES', 1);
define('POS_ANIO', 2);
define('SEPARADOR', ';');
define('SEPARADOR_FECHA', '/');
define('ENCABEZADO_SALIDA', "Estacion;Año;Mes;Tipo;Promedio\n");

function leer_estructura($nombre_archivo=ARCHIVO_DATOS){
    $archivo = fopen($nombre_archivo, 'r');
    $datos_climaticos = array();
    if ($archivo){
        fgets($archivo); //Se ignora el encabezado
        while (!feof($archivo)){
            $reg = explode(SEPARADOR, fgets($archivo));
            $fecha = explode (SEPARADOR_FECHA, $reg[POS_FECHA]);
            $datos_climaticos[$reg[POS_ESTACION]][$fecha[POS_ANIO]][$fecha[POS_MES]][$reg[POS_TIPO]][] = $reg[POS_VALOR];
        }
        fclose($archivo);
    }
    return $datos_climaticos;
}

function promedio($arreglo){
    $promedio = 0;
    $suma = array_sum($arreglo);
    $cantidad = count($arreglo);
    if ($cantidad > 0)
        $promedio = $suma / $cantidad;
    return $promedio;
}

function escribir_promedios($promedios, $nombre_archivo=ARCHIVO_SALIDA){
    $archivo = fopen($nombre_archivo, 'w+');
    if ($archivo){
        fputs($archivo, ENCABEZADO_SALIDA);
        foreach ($promedios as $estacion => $datos_sensores)
            foreach ($datos_sensores as $anio => $datos_anuales)
                foreach ($datos_anuales as $mes => $datos_mensuales)
                    foreach ($datos_mensuales as $tipo => $valores)
```

```
fputs($archivo,$estacion . SEPARADOR . $anio . SEPARADOR . $mes .  
SEPARADOR . $tipo . SEPARADOR . promedio($valores) . "\n");  
fclose($archivo);  
}  
}  
escribir_promedios(Leer_Estructura());  
?>
```

## Una alternativa de resolución del Parcial de la cursada 2012

### Enunciado

#### Enunciado del Parcial

```
<?php  
define('ARCHIVO_DATOS' , 'datos.txt');  
define('ARCHIVO_SALIDA_SUSPENSIONES', 'suspensiones.txt');  
define('ARCHIVO_SALIDA_ESTADISTICA', 'estadistica_infracciones.txt');  
define('CANTIDAD_SUSPENSION', 30);  
define('POS_INFRACTOR', 0);  
define('POS_APYNO', 1);  
define('POS_COD_INFR', 2);  
define('POS_PUNTOS', 3);  
define('SEPARADOR', ';');  
  
function leer_estructura($nombre_archivo=ARCHIVO_DATOS){  
    $archivo = fopen($nombre_archivo,'r');  
    $datos_infractores =  
array('infractores'=>array(), 'por_infraccion'=>array());  
    if ($archivo){  
        fgets($archivo); //Se ignora el encabezado  
        while (!feof($archivo)){  
            $reg = explode(SEPARADOR,trim(fgets($archivo)));  
            $datos_infractores['infractores'][$reg[POS_INFRACTOR]]['nombre']=$reg[POS_APYNO];  
            $datos_infractores['infractores'][$reg[POS_INFRACTOR]]['infracciones']+=$reg[POS_PUNTOS];  
            $datos_infractores['por_infraccion'][$reg[POS_COD_INFR]]['total']+=$reg[POS_PUNTOS];  
            $datos_infractores['por_infraccion'][$reg[POS_COD_INFR]]['cantidad']++;  
        }  
    }  
    return $datos_infractores;  
}  
  
function escribir_suspensiones($valores,  
$nombre_archivo=ARCHIVO_SALIDA_SUSPENSIONES){  
    $archivo = fopen($nombre_archivo,'w+');
```

```

    if ($archivo){
        fputs($archivo,"Cod. Infractor;Nombre y Apellido; Puntos\n");
        foreach ($valores['infractores'] as $infractor => $datos_infractor)
            if ($datos_infractor['infracciones']>=CANTIDAD_SUSPENSION)
                fputs($archivo,$infractor.SEPARADOR.$datos_infractor['nombre'].SEPARADOR.$datos_infractor['infracciones']."\n");
        fclose($archivo);
    }
}

function escribir_estadistica($valores,
$nombre_archivo=ARCHIVO_SALIDA_ESTADISTICA){
    $archivo = fopen($nombre_archivo,'w+');
    if ($archivo){
        fputs($archivo,"Cod. Infracción;Puntos;Total de Infracciones\n");
        foreach ($valores['por_infraccion'] as $infraccion => $datos_infraccion)
            fputs($archivo,$infraccion.SEPARADOR.$datos_infraccion['total'].SEPARADOR.$datos_infraccion['cantidad']."\n");
        fclose($archivo);
    }
}

$datos_infractores = leer_estructura();
escribir_suspensiones($datos_infractores);
escribir_estadistica($datos_infractores);
?>

```

## Una alternativa de resolución del Prefinal de la cursada 2010

### Enunciado

Enunciado del prefinal Resolución en Formato Libre Office/Open Office

```

<?php
define('ARCHIVO_DATOS' , 'datos.txt');
define('ARCHIVO_SALIDA', 'salida.txt');
define('POS_HC',0); // HC Significa Historia Clínica
define('POS_FECHA',1);
define('POS_TIPO' ,3);
define('POS_VALOR',4);
define('SEPARADOR',';');
define('ENCABEZADO_SALIDA',"Historia Clinica;Fecha;Tipo;Valor\n");

function leer_estructura($nombre_archivo=ARCHIVO_DATOS){
    $archivo = fopen($nombre_archivo,'r');
    $datos_paciente = array();
    if ($archivo){
        fgets($archivo); //Se ignora el encabezado
        while (!feof($archivo)){
            $reg = explode(SEPARADOR,trim(fgets($archivo)));
            $datos_paciente[$reg[POS_HC]][$reg[POS_FECHA]][$reg[POS_TIPO]][] =

```

```
$reg[POS_VALOR];  
    }  
}  
return $datos_paciente;  
}  
function escribir_valores($valores, $nombre_archivo=ARCHIVO_SALIDA){  
    $archivo = fopen($nombre_archivo,'w+');  
    if ($archivo){  
        fputs($archivo,ENCABEZADO_SALIDA);  
        foreach ($valores as $paciente => $datos_vitales)  
            foreach ($datos_vitales as $dia => $datos_diarios)  
                foreach ($datos_diarios as $tipo => $datos){  
                    fputs($archivo,$paciente . SEPARADOR . $dia . SEPARADOR . "$tipo+" .  
SEPARADOR . max($datos) . "\n"); //función max nativa de PHP  
                    fputs($archivo,$paciente . SEPARADOR . $dia . SEPARADOR . "$tipo-" .  
SEPARADOR . min($datos) . "\n"); //función min nativa de PHP  
                }  
            fclose($archivo);  
        }  
}  
escribir_valores(Leer_Estructura());  
?>
```

[parcialphp2010.pdf](#)

## Parcial del 2011

### Enunciado

[Enunciado del parcial](#)

### Una alternativa de Resolución

```
<?php  
define ('ARCHIVO_ENTRADA','datos.txt');  
define ('ARCHIVO_SALIDA','consumosmenores.txt');  
define ('SEPARADOR',';');  
define ('POS_NOMBRE',0);  
define ('POS_DETALLE',1);  
define ('POS_CONSUMO',2);  
define ('IMPORTE_BASE',50);  
  
function leer_consumos($nombre_archivo=ARCHIVO_ENTRADA){  
    $archivo = fopen($nombre_archivo,'r');  
    $datos_consumo = array();
```



```

if ($archivo){
    while (!feof($archivo)){
        $reg = explode(SEPARADOR,trim(fgets($archivo)));
        if (!isset($datos_consumo[$reg[POS_NOMBRE]]))
            $datos_consumo[$reg[POS_NOMBRE]]['total'] = 0;
        $datos_consumo[$reg[POS_NOMBRE]]['total'] += $reg[POS_CONSUMO];
    }
}
return $datos_consumo;
}

function
escribir_consumos($datos_consumo,$nombre_archivo_entrada=ARCHIVO_ENTRADA,$no
mbre_archivo_salida=ARCHIVO_SALIDA, $max_consumo=IMPORTE_BASE){
    $archivo_entrada = fopen($nombre_archivo_entrada,'r');
    $archivo_salida = fopen($nombre_archivo_salida,'w');
    if ($archivo_entrada and $archivo_salida){
        while (!feof($archivo_entrada)){
            $reg = explode(SEPARADOR,trim(fgets($archivo_entrada)));
            if ($datos_consumo[$reg[POS_NOMBRE]]['total']<$max_consumo)
                fputs($archivo_salida,implode(SEPARADOR,$reg)."\n");
        }
        fclose($archivo_entrada);
        fclose($archivo_salida);
    }
}
escribir_consumos(Leer_consumos());
?>

```

## Recuperatorio del 2011

### Una alternativa de Resolución

```

<?php
define ('DIAS_ANIO',365);
define ('MESES',12);
define ('ARCHIVO_SALIDA','historicos.txt');
$diasmes = array(
    1 => array('dias' =>31,'nombre'=>'Enero'),
    2 => array('dias' =>28,'nombre'=>'Febrero'),
    3 => array('dias' =>31,'nombre'=>'Marzo'),
    4 => array('dias' =>30,'nombre'=>'Abril'),
    5 => array('dias' =>31,'nombre'=>'Mayo'),
    6 => array('dias' =>30,'nombre'=>'Junio'),
    7 => array('dias' =>31,'nombre'=>'Julio'),
    8 => array('dias' =>31,'nombre'=>'Agosto'),
    9 => array('dias' =>30,'nombre'=>'Septiembre'),
    10 => array('dias' =>31,'nombre'=>'Octubre'),

```

```
11 => array('dias' =>30,'nombre'=>'Noviembre'),
12 => array('dias' =>31,'nombre'=>'Diciembre')
);
/*
 * Esta función se implementa para poder poblar el arreglo inicial y probar
el ejercicio
 * No es necesario para la resolución
 */
function inicializar_precipitaciones(){
    $precipitaciones = array();
    for ($i=0; $i<DIAS_ANIO;$i++)
        $precipitaciones[] = rand(0,50);
    return $precipitaciones;
}
function agrupar_precipitaciones($precipitaciones,$diasmes){
    $matrizprecipitaciones = array();
    $cantidad = count($precipitaciones);
    $posactual = 0;
    $posmax = 0;
    for ($mesactual=1; $mesactual<=MESES; $mesactual++){
        $posmax += $diasmes[$mesactual]['dias'];
        for ($dia=$posactual;$dia<$posmax;$dia++){
            if ($precipitaciones[$dia]>0)
                $matrizprecipitaciones[$diasmes[$mesactual]['nombre']][$dia-
$posactual+1]=$precipitaciones[$dia];
        }
        $posactual = $posmax;
    }
    return $matrizprecipitaciones;
}
function promedio($arreglo,$dias){
    $total = 0;
    foreach($arreglo as $dia=>$precip)
        $total += $precip;
    return $total / $dias;
}
function armar_promedios($matrizprecipitaciones,$diasmes){
    $historicos = array();
    for ($mesactual=1; $mesactual<=MESES; $mesactual++)
        $historicos[$diasmes[$mesactual]['nombre']] =
promedio($matrizprecipitaciones[$diasmes[$mesactual]['nombre']], $diasmes[$me
sactual]['dias']);
    /*
     * Se emplea la función array_multisort de PHP. Se asume el
conocimiento de la misma
     * En caso de no conocerla y definir el encabezado de la función se
consideraba correcto
     * No era necesaria su implementación
     */
}
```

```

        array_multisort($historicos, SORT_DESC);
        return $historicos;
    }
    function crear_archivo($historicos, $nombre_archivo_salida =
    ARCHIVO_SALIDA){
        $archivo_salida = fopen($nombre_archivo_salida, 'a+');
        if ($archivo_salida){
            foreach($historicos as $mes => $promedio){
                fwrite($archivo_salida, "$mes\t$promedio\n");
            }
            fclose($archivo_salida);
        }
    }
    $precipitaciones = inicializar_precipitaciones();
    $matriz_precipitaciones =
    agrupar_precipitaciones($precipitaciones, $diasmes);
    $historicos = armar_promedios($matriz_precipitaciones, $diasmes);
    crear_archivo($historicos);
    ?>

```

## Prefinal del 2011

### Una alternativa de Resolución

```

<?php
define('ARCHIVO_DATOS' , 'datos.txt');
define('ARCHIVO_SALIDA', 'salida.txt');
define('POS_ESTACION', 0);
define('POS_FECHA', 1);
define('POS_TIPO', 3);
define('POS_VALOR', 4);
define('SEPARADOR', ';');
define('ENCABEZADO_SALIDA', "Estacion;Fecha;Tipo;Promedio\n");

function leer_estructura($nombre_archivo=ARCHIVO_DATOS){
    $archivo = fopen($nombre_archivo, 'r');
    $datos_climaticos = array();
    if ($archivo){
        fgets($archivo); //Se ignora el encabezado
        while (!feof($archivo)){
            $reg = explode(SEPARADOR, fgets($archivo));
            $datos_climaticos[$reg[POS_ESTACION]][$reg[POS_FECHA]][$reg[POS_TIPO]][] =
            $reg[POS_VALOR];
        }
    }
    return $datos_climaticos;
}

function promedio($arreglo){

```

```
$promedio = 0;
$suma = 0;
$cantidad = count($arreglo);
if ($cantidad > 0){
    foreach ($arreglo as $val)
        $suma += $val;
    $promedio = $suma / $cantidad;
}
return $promedio;
}

function calcular_promedios_por_tipo($datos_climaticos){
    $promedios = array();
    foreach ($datos_climaticos as $estacion => $datos_sensores)
        foreach ($datos_sensores as $dia => $datos)
            foreach ($datos as $tipo => $valores)
                $promedios[$estacion][$dia][$tipo] = promedio($valores);
    return $promedios;
}

function escribir_promedios($promedios, $nombre_archivo=ARCHIVO_SALIDA){
    $archivo = fopen($nombre_archivo, 'w+');
    if ($archivo){
        fputs($archivo, ENCABEZADO_SALIDA);
        foreach ($promedios as $estacion => $datos_sensores)
            foreach ($datos_sensores as $dia => $datos)
                foreach ($datos as $tipo => $valor)
                    fputs($archivo, $estacion . SEPARADOR . $dia . SEPARADOR . $tipo .
SEPARADOR . $valor . "\n");
        fclose($archivo);
    }
}

$datos_climaticos = leer_estructura();
$promedios = calcular_promedios_por_tipo($datos_climaticos);
escribir_promedios($promedios);
?>
```

## Estilo de enunciado similar al Parcial que se puede tomar

### Enunciado

[Enunciado del parcial](#)

### Una alternativa de Resolución

```
<?php
define('SEPARADOR_CAMPOS', ';');
```

```
define('POS_CAMPO_PRODUCTO', 0);
define('POS_CAMPO_FECHA', 1); // en este problema en particular pareciera
que no se usa...
define('POS_CAMPO_PRECIO', 2);

/**
 * Carga el contenido del archivo de nombre $nombre_entrada y lo devuelve
 como un arreglo asociativo.
 *
 * @param string $nombre_entrada nombre del archivo que contiene los precios
 * @return mixed un arreglo con los precios, si el archivo se cargó3
 correctamente;
 *
 *         false si hubo algún error
 */
function precios_desde_archivo($nombre_entrada) {
    $precios = false;
    $archivo_precios = fopen($nombre_entrada, 'r');
    if ($archivo_precios) {
        $precios = array();
        while (!feof($archivo_precios)) {
            $linea = fgets($archivo_precios);
            // Si el archivo tiene un \n en la última línea, en la última
vuelta
            // feof devuelve false (entra al while), pero fgets no tiene una
línea
            // para leer y devuelve false también
            if (!feof($archivo_precios)) {
                $campos = explode(SEPARADOR_CAMPOS, trim($linea));
                if (!array_key_exists($campos[POS_CAMPO_PRODUCTO],
$precios)) {
                    $precios[$campos[POS_CAMPO_PRODUCTO]] = array(
                        'inicial' => $campos[POS_CAMPO_PRECIO],
                        'final' => $campos[POS_CAMPO_PRECIO]
                    );
                } else {
                    $precios[$campos[POS_CAMPO_PRODUCTO]]['final'] =
$campos[POS_CAMPO_PRECIO];
                }
            }
            fclose($archivo_precios);
        }
        return $precios;
    }
}

/**
 * Calcula la diferencia porcentual entre dos valores.
 *
 * @param float $valor_inicial
 * @param float $valor_final
 * @return float
```

```
*/  
function diferencia_porcentual($valor_inicial, $valor_final) {  
    return (($valor_final / $valor_inicial) - 1) * 100;  
}  
  
/**  
 * Agrega a cada producto en $precios la diferencia porcentual entre el  
 * precio inicial y el final.  
 *  
 * @param array $precios datos sobre los precios de los productos  
 */  
function calcular_variaciones_precios(&$precios) {  
    foreach ($precios as $producto => $datos_producto) {  
        $precios[$producto]['diferencia'] =  
diferencia_porcentual($precios[$producto]['inicial'],  
$precios[$producto]['final']);  
    }  
}  
  
/**  
 * Devuelve estadísticas sobre los precios dados.  
 *  
 * @param array $precios arreglo con los precios de los productos  
 * @return array un arreglo con las claves:  
 *  
 *      - 'maximo': máxima diferencia de precio  
 *      - 'producto_maximo': producto con la máxima diferencia de  
precio  
 *      - 'minimo': mínima diferencia de precio  
 *      - 'producto_minimo': producto con la mínima diferencia de  
precio  
 *      - 'promedio': diferencia de precio promedio entre todos los  
productos  
 */  
function estadisticas_variacion_precios($precios) {  
    $estadisticas = array();  
    $productos = array_keys($precios);  
    $max = $min = $dif_total = $precios[$productos[0]]['diferencia'];  
    $prod_max = $prod_min = $productos[0];  
    for ($i = 1; $i < count($precios); $i++) {  
        $diferencia = $precios[$productos[$i]]['diferencia'];  
        if ($diferencia > $max) {  
            $max = $diferencia;  
            $prod_max = $productos[$i];  
        }  
        if ($diferencia < $min) {  
            $min = $diferencia;  
            $prod_min = $productos[$i];  
        }  
        $dif_total += $diferencia;  
    }  
}
```

```

    }
    $estadisticas['maximo'] = $max;
    $estadisticas['producto_maximo'] = $prod_max;
    $estadisticas['minimo'] = $min;
    $estadisticas['producto_minimo'] = $prod_min;
    $estadisticas['promedio'] = $dif_total / count($precios);
    return $estadisticas;
}

/**
 * Guarda en el archivo de nombre $nombre_salida los datos dados en
 * $variaciones.
 *
 * @param string $nombre_salida nombre del archivo de salida
 * @param array $variaciones datos a guardar, con el formato devuelto por
 * estadisticas_variacion_precios
 * @return boolean true si se la operación se completó correctamente; false
 * si hubo algún problema
 */
function guardar_variaciones_precios($nombre_salida, $variaciones) {
    $archivo_variaciones = fopen($nombre_salida, 'a');
    if ($archivo_variaciones) {
        $salida = "Máximo" . SEPARADOR_CAMPOS .
$variaciones['producto_maximo'] . SEPARADOR_CAMPOS . $variaciones['maximo']
. "\n";
        $salida .= "Mínimo" . SEPARADOR_CAMPOS .
$variaciones['producto_minimo'] . SEPARADOR_CAMPOS . $variaciones['minimo']
. "\n";
        $salida .= "Promedio" . SEPARADOR_CAMPOS . "Todos" .
SEPARADOR_CAMPOS . $variaciones['promedio'] . "\n";
        $ok = fwrite($archivo_variaciones, $salida);
        $resultado = $ok !== false;
        fclose($archivo_variaciones);
    } else
        $resultado = false;
    return $resultado;
}

define('ARCHIVO_PRECIOS', 'precios.txt');
define('ARCHIVO_VARIACIONES', 'variaciones.txt');

$precios = precios_desde_archivo(ARCHIVO_PRECIOS);
if ($precios !== false) { // debe usarse !==, porque también puede retornar
un array vacío, que es válido
    if (count($precios) > 0) { // si el archivo está vacío, no hay más para
hacer
        calcular_variaciones_precios($precios);
        $estadisticas = estadisticas_variacion_precios($precios);
        $ok = guardar_variaciones_precios(ARCHIVO_VARIACIONES,
$estadisticas);
        if ($ok)

```

```
        echo "Se guardaron correctamente las variaciones de precios.";
    else
        echo "Hubo un error al guardar el archivo de variaciones (" .
ARCHIVO_VARIACIONES . ").\n";
    } else
        echo "El archivo " . ARCHIVO_VARIACIONES . " no contiene precios
guardados.\n";
    } else {
        echo "Hubo un error al leer el archivo de precios (" . ARCHIVO_PRECIOS .
").\n";
    }
    ?>
```

From:  
<https://wiki.rec.unicen.edu.ar/wiki/> - Wiki UNICEN

Permanent link:  
<https://wiki.rec.unicen.edu.ar/wiki/doku.php?id=programacionphp2:exámenes:start&rev=1342190712>

Last update: 2017/10/10 16:08

