

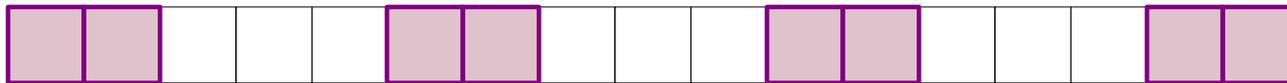
Tipos de Datos Derivados Regulares: Ejemplos

```
MPI_Type_vector(count, length, stride, type, newtype)
```

Crea un tipo de datos homogéneo y regular a partir de elementos de un *array* equiespaciados

- 1 De cuántos bloques se compone (`count`)
- 2 De qué longitud son los bloques (`length`)
- 3 Qué separación hay entre un elemento de un bloque y el mismo elemento del siguiente bloque (`stride`)
- 4 De qué tipo son los elementos individuales (`type`)

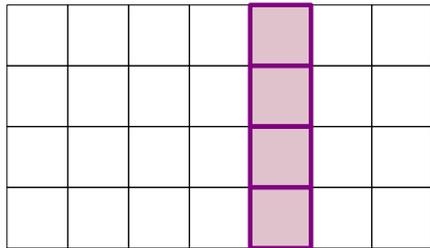
Ejemplo: tipo de datos necesario para enviar/recibir los elementos coloreados (de un vector de enteros)



```
MPI_Datatype tipo;  
MPI_Type_vector(4,2,5,MPI_INT,&tipo);
```

Tipos de Datos Derivados Regulares: Ejemplos

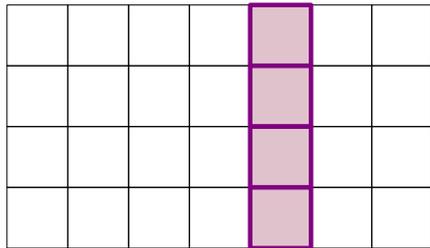
- Sobre el ejemplo de la transparencia 46, modificar el código para enviar/recibir la columna 4 en vez de la 2



```
double A[4][7];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) { /* envía la 3a columna */
    MPI_Send(&A[0][2], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, columna, 0, 0, comm, &status);
}
```

Tipos de Datos Derivados Regulares: Ejemplos

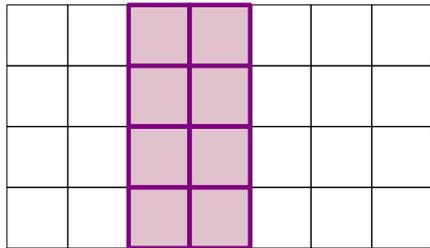
- Sobre el ejemplo de la transparencia 46, modificar el código para enviar/recibir la columna 4 en vez de la 2



```
double A[4][7];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) { /* envía la 3a columna */
    MPI_Send(&A[0][4], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(&A[0][4], 1, columna, 0, 0, comm, &status);
}
```

Tipos de Datos Derivados Regulares: Ejemplos

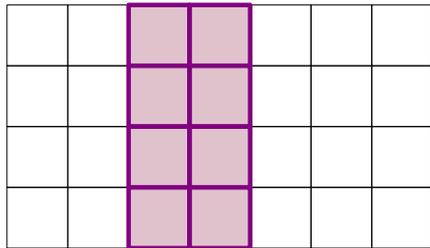
- Sobre el ejemplo de la transparencia 46, modificar el código para enviar/recibir el bloque de las columnas 2 y 3



```
double A[4][7];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) { /* envía la 3a columna */
    MPI_Send(&A[0][2], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, columna, 0, 0, comm, &status);
}
```

Tipos de Datos Derivados Regulares: Ejemplos

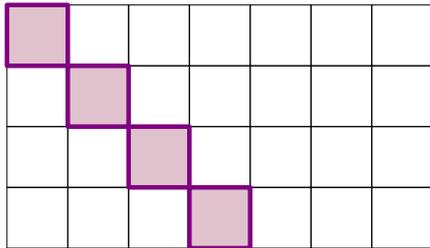
- Sobre el ejemplo de la transparencia 46, modificar el código para enviar/recibir el bloque de las columnas 2 y 3



```
double A[4][7];
MPI_Datatype bloque;
MPI_Type_vector(4, 2, 7, MPI_DOUBLE, &bloque);
MPI_Type_commit(&bloque);
if (my_rank == 0) { /* envía la 3a columna */
    MPI_Send(&A[0][2], 1, bloque, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, bloque, 0, 0, comm, &status);
}
```

Tipos de Datos Derivados Regulares: Ejemplos

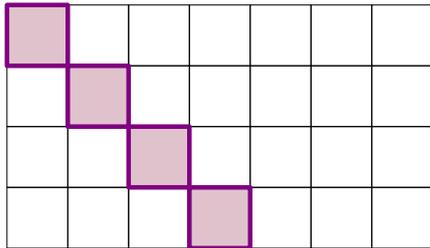
- Sobre el ejemplo de la transparencia 46, modificar el código para enviar/recibir los elementos de la diagonal



```
double A[4][7];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) { /* envía la 3a columna */
    MPI_Send(&A[0][2], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, columna, 0, 0, comm, &status);
}
```

Tipos de Datos Derivados Regulares: Ejemplos

- Sobre el ejemplo de la transparencia 46, modificar el código para enviar/recibir los elementos de la diagonal



```
double A[4][7];
MPI_Datatype diag;
MPI_Type_vector(4, 1, 8, MPI_DOUBLE, &diag);
MPI_Type_commit(&diag);
if (my_rank == 0) { /* envía la 3a columna */
    MPI_Send(&A[0][0], 1, diag, 1, 0, comm);
} else {
    MPI_Recv(&A[0][0], 1, diag, 0, 0, comm, &status);
}
```

Tipos de Datos Derivados Regulares: Ejemplos

- Sobre el ejemplo de la transparencia 46, modificar el código para enviar la columna 2, pero recibiendo los elementos de forma contigua sobre un array `double b[4]`

```
double A[4][7];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) { /* envía la 3a columna */
    MPI_Send(&A[0][2], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, columna, 0, 0, comm, &status);
}
```

Tipos de Datos Derivados Regulares: Ejemplos

- Sobre el ejemplo de la transparencia 46, modificar el código para enviar la columna 2, pero recibiendo los elementos de forma contigua sobre un array `double b[4]`

```
double A[4][7], b[4];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) { /* envía la 3a columna */
    MPI_Send(&A[0][2], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(b, 4, MPI_DOUBLE, 0, 0, comm, &status);
}
```

Tipos de Datos Derivados Regulares: Ejemplos

- Sobre el ejemplo de la transparencia 46, modificar el código para enviar un array `double b[4]`, recibiendo los elementos sobre la columna 2 de la matriz A

```
double A[4][7];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) { /* envía la 3a columna */
    MPI_Send(&A[0][2], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, columna, 0, 0, comm, &status);
}
```

Tipos de Datos Derivados Regulares: Ejemplos

- Sobre el ejemplo de la transparencia 46, modificar el código para enviar un array `double b[4]`, recibiendo los elementos sobre la columna 2 de la matriz A

```
double A[4][7], b[4];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) {                /* envía la 3a columna */
    MPI_Send(b, 4, MPI_DOUBLE, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, columna, 0, 0, comm, &status);
}
```