

Reunió Social

Enunciat

Volem fer un programa per tal de simular una reunió social (o festa). L'objectiu és bàsicament veure com es mouen dins l'escenari les persones que formen part de la festa en funció de diversos interessos.

L'ESCENARI

Considerarem una sala rectangular. Caldrà donar, com a dades d'entrada, l'amplada i llargada (parlant en termes de matrius, el nombre de files i columnes).

ELS PARTICIPANTS

A la reunió social hi haurà d'una banda els cambrers i de l'altra els convidats, que podran ser homes o dones.

LA SIMULACIÓ

Els integrants de la festa han d'anar-se movent per l'escenari a partir de les següents regles bàsiques:

- A partir d'una posició, una persona pot quedar-se allà mateix o bé anar a les 4 posicions adjacents (dalt, baix, dreta i esquerra)
- En una posició no pot haver més d'una persona.
- No està permès sortir fora dels límits de l'escenari.
- Totes les persones han de tenir les mateixes oportunitats de moure's. Això vol dir que, per a cada iteració, una persona només es podrà moure 1 cop.

Els interessos de les persones també hauran de ser considerats:

- a) **Cambrers**: els seu interès és que els convidats estiguin ben atesos, per tant tendiran a moure's cap a on hi ha convidats (sense fer cap distinció entre ells) i tendiran a allunyar-se dels altres cambrers.
- b) **Convidats**: en general, tendeixen a apropar-se als convidats que els hi cauen més bé i a allunyar-se dels que els cauen pitjor. A més, el fet de ser de diferent sexe sol ser un motiu addicional "d'atracció" (tot i que no sempre). També considerarem que els homes, a diferència de les dones, tenen una certa tendència a anar cap a on hi ha cambrers.

MODELITZACIÓ

Cal donar valor numèric a tot plegat i fer un model. Definirem el concepte d'atracció A d'un individu k cap a una posició (f,c) de l'escenari:

$$A_{(f,c)}^k = \sum_{(i=0, j=0)}^{(Files, Cols)} \frac{\sigma_{i,j}^k}{\sqrt{(i-f)^2 + (j-c)^2}}$$

on el sumatori és sobre totes les posicions (i,j) de l'escenari excepte la mateixa posició (f,c) (d'aquesta manera evitem la divisió per zero).

El denominador de la fracció representa la distància entre la posició (f,c) i la posició (i,j) (Ha de ser diferent de zero, i per tant, no considerem la mateixa casella, que estaria a distància zero). El numerador representa un factor d'interès de l'individu k cap al que hi ha a la posició (i,j) , sigui el que sigui (o sigui un cambrer, un convidat o bé res).

Cal establir ara el factor d'interès $\sigma_{i,j}^k$. Si a (i,j) no hi ha res, l'interès serà 0 en qualsevol cas. Això també passa en cas que hi hagi el mateix individu k . Altrament s'haurà de diferenciar:

En cas que k sigui un cambrer, considerarem un interès positiu (+1) si hi ha un convidat, i negatiu (-1) si hi ha un altre cambrer.

Si k és un convidat, caldrà tenir en compte la seva afinitat (simpatia) envers la resta de convidats (no envers els cambrers). Aquesta afinitat vindrà representada per un valor enter entre -5 i 5 (-5 voldrà dir rebuig total, 0 indiferència i 5 afinitat total). Així doncs cada convidat haurà de tenir la seva taula d'afinitats, i per tant l'interès d'un convidat envers una posició, en cas que aquesta estigui ocupada per un

altre convidat, ve donat per aquesta taula. Però també cal considerar (sumar), si és el cas, el plus d'interès per convidats de sexe contrari (que depèn de cada convidat).
Pel que fa a l'interès dels convidats respecte els cambrers, considerarem que existeix en cas de que el convidat sigui un home. Podem representar aquest interès amb un valor de +1 (el mateix per a tots els homes).

DADES D'ENTRADA

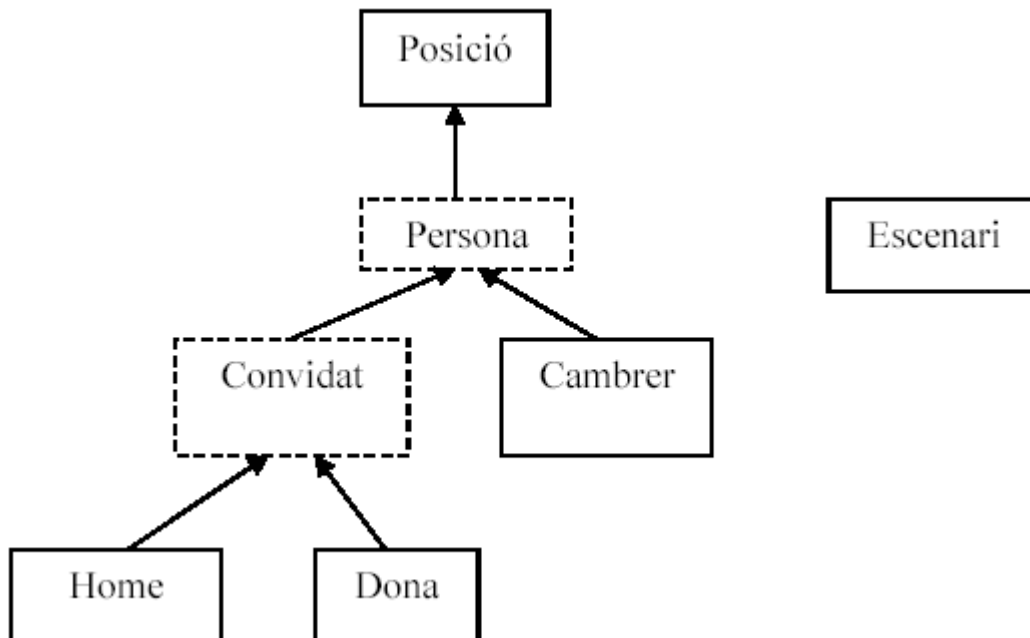
- Nombre de files i columnes de l'escenari (cal establir uns màxims adequats a les dimensions de la pantalla).
 - Nombre de cambrers (màxim 20). Els cambrers vindran representats en pantalla mitjançant un asterisc '*'.
 - Nombre d'homes i dones (posem un màxim de 100 convidats en total).
 - Per cada convidat
 - o Nom: Símbol amb que el volem representar a pantalla (per exemple inicial del nom, etc.).
 - o Plus d'interès per convidats del sexe contrari (0..3)
 - o Afinitat amb cadascú de la resta dels convidats
- La posició inicial de convidats i cambrers serà donada de manera aleatòria.

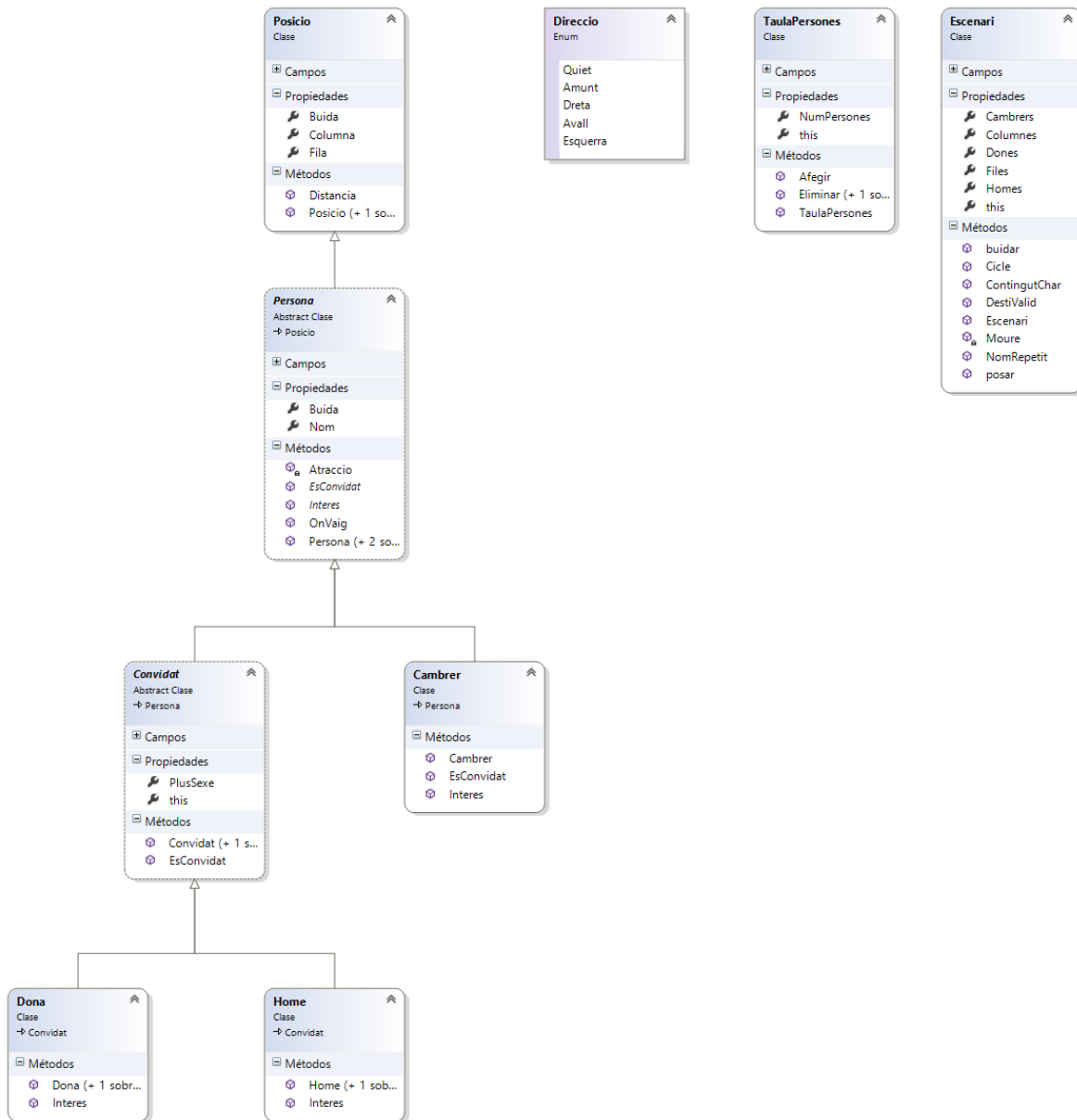
SORTIDA

Volem que aparegui, en pantalla, la representació de l'escenari de la festa on s'identifiqui els cambrers i els diferents convidats, de manera que es pugui veure amb claredat com van movent-se per l'escenari al llarg de la festa.

Implementació

El diagrama de classes serà el següent:





Les classes que cal implementar tenen la interfície següent:

Classe Posicio

```
public class Posicio
{
    /// <summary>
    /// Crea una nova posició
    /// </summary>
    /// <param name="fil">Fila de la Posició</param>
    /// <param name="col">Columna de la Posició</param>
    public Posicio(int fil, int col)

    /// <summary>
    /// Crea una nova posició amb fila i columna igual a 0
    /// </summary>
    public Posicio()

    /// <summary>
    /// Assigna o obté la columna de la posicio
    /// </summary>
    public int Columna
    {
        get;
        set;
    }

    /// <summary>
    /// Assigna o obté la fila de la posicio
    /// </summary>
    public int Fila
    {
        get ;
        set ;
    }

    /// <summary>
    /// Retorna si la posició està o no buida
    /// </summary>
    public virtual bool Buida
    {
        get ;
    }

    /// <summary>
    /// Retorna la distància entre dues posicions
    /// </summary>
    /// <param name="pos1">Primera posició</param>
    /// <param name="pos2">Segona posició</param>
    /// <returns>Distància entre les dues posicions</returns>
    public static double Distancia (Posicio pos1, Posicio pos2)
```

Classe Persona

```
public abstract class Persona:Posicio
{
    /// <summary>
    /// Crea una persona
    /// </summary>
    /// <param name="nom">Strng que identifica la persona</param>
    /// <param name="fil">Fila on està localitzada</param>
    /// <param name="col">Columna on està localitzada</param>
    public Persona(string nom, int fil, int col):base(fil, col)

    /// <summary>
    /// Crea una persona
    /// </summary>
    /// <param name="nom">nom de la persona</param>
    public Persona(string nom)

    /// <summary>
    /// Crea una persona
    /// </summary>
    public Persona ():base()

    /// <summary>
    /// Obté el nom de la persona
    /// </summary>
    public string Nom
    {
        get ;
    }
    /// <summary>
    /// Retorna que la posició ocupada per aquesta persona no està buida
    /// </summary>
    public override bool Buida
    {
        get ;
    }

    /// <summary>
    /// Atracció de persona sobre una determinada posicio
    /// </summary>
    /// <param name="fil">Fila de la posició</param>
    /// <param name="col">Columan de la posició</param>
    /// <param name="esc">Escenari on estan situats</param>
    /// <returns>Atracció quantificada</returns>
    private double Atraccio (int fil, int col, Escenari esc)

    /// <summary>
    /// Decideix quin serà el següent moviment que farà la persona
    /// </summary>
    /// <param name="esc">Escenari on esta situada la persona</param>
    /// <returns>Una de les 5 possibles direccions (Quiet, Amunt, Avall, Dreta, Esquerra</returns>
    public Direccio OnVaig (Escenari esc)
```

```
/// <summary>
/// Interès de la persona sobre una determinada posició
/// </summary>
/// <param name="pos">Posició</param>
/// <returns>Interès quantificat</returns>
public abstract int Interes (Posicio pos);

/// <summary>
/// Determina si la persona es un convidat (home o dona) o un cambrer
/// </summary>
/// <returns>Retorna si és convidat</returns>
public abstract bool EsConvidat();
}
```

Classe Convidat

```
public abstract class Convidat:Persona
{
    /// <summary>
    /// Crea un convidat
    /// </summary>
    /// <param name="nom">string que l'identificarà</param>
    /// <param name="simp">Taula de simpaties</param>
    /// <param name="sex">Plus de simpatia sobre el sexe contrari</param>
    public Convidat(string nom,int[] simp, int sexe)

    /// <summary>
    /// Crea un convidat
    /// </summary>
    /// <param name="nom">Caràcter que l'identificarà</param>
    /// <param name="sex">Plus de simpatia sobre el sexe contrari</param>
    public Convidat(string nom,int sexe)

    /// <summary>
    /// Retorna o estableix la simpaties envers a algú
    /// </summary>
    public int this [string nom]
    {
        get ;
        set ;
    }

    /// <summary>
    /// Retorna o estableix el plus de simpatia envers del sexe contrari
    /// </summary>
    public int PlusSexe
    {
        get ;
        set ;
    }

    /// <summary>
    /// Retorna que si és un convidat
    /// </summary>
    /// <returns>Cert</returns>
    public override bool EsConvidat()
}
```

Classe Dona

```
public class Dona:Convidat
{

    /// <summary>
    /// Crea una Dona
    /// </summary>
    /// <param name="nom">String que identifica aquesta Dona</param>
    /// <param name="simpa">Taula de simpaties envers els altres convidats</param>
    /// <param name="sexe">Plus de simpatia envers convidats del sexe contrari</para
    m>
    public Dona(string nom, int[] simpa,int sexe):base(nom,simpa,sexe)

    /// <summary>
    /// Crea una Dona
    /// </summary>
    /// <param name="nom"> String que identifica aquesta Dona</param>
    /// <param name="sexe">Plus de simpatia envers convidats del sexe contrari</para
    m>
    public Dona(string nom, int sexe)

    /// <summary>
    /// Interès d'aquest home per una posició
    /// </summary>
    /// <param name="pos">Posició per la qual s'interessa</param>
    /// <returns>Interès quantificat</returns>
    public override int Interes(Posicio pos)

}
```


Classe Home

```
public class Home:Convidat
{

    /// <summary>
    /// Crea un Home
    /// </summary>
    /// <param name="nom">String que l'identificarà</param>
    /// <param name="simpa">Taula de simpaties</param>
    /// <param name="sexe">Plus de simpatia envers del sexe contrari</param>
    public Home(string nom,  int []simpa, int sexe

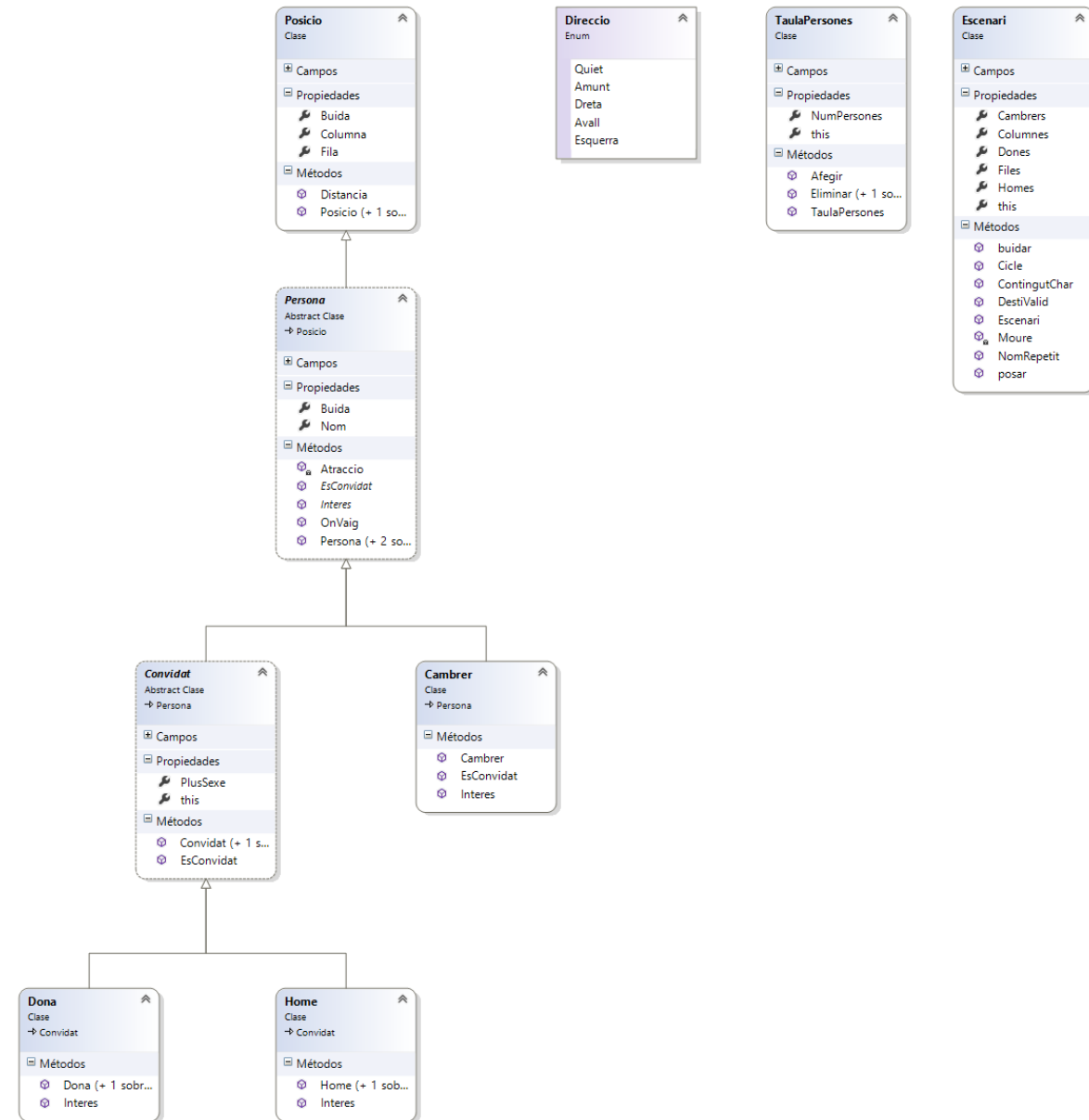
    /// <summary>
    /// Crea un Home
    /// </summary>
    /// <param name="nom">String que l'identificarà</param>
    /// <param name="sexe">Plus de simpatia envers del sexe contrari</param>
    public Home(string nom,  int sexe)

    /// <summary>
    /// Interès d'aquest home per una posició
    /// </summary>
    /// <param name="pos">Posició per la qual s'interessa</param>
    /// <returns>Interès quantificat</returns>
    public override int Interes(Posicio pos)
}
```

Classe Cambrer

```
public class Cambrer:Persona
```

```
{
  /// <summary>
  /// Crea un cambrer (Persona de la que no importa el nom, i es dirà "Cambrer 1",
  /// "Cambrer 2", "Cambrer 3", "Cambrer 4" ... "Cambrer
  N
```



```
"/// </summary>
```

```
public Cambrer()
```

```
/// <summary>
```

```
/// Interès del cambrer per una posició
```

```
/// </summary>
```

```
/// <param name="pos">posició per la que s'interessa</param>
```

```
/// <returns>Retorna 0 si no hi ha ningú, 1 si hi ha un convidat i -  
1 si un cambrer</returns>  
public override int Interes(Posicio pos)  
  
/// <summary>  
/// Retorna que el Cambrer no és un convidat  
/// </summary>  
/// <returns>false</returns>  
public override bool EsConvidat()  
}
```

Enumeració Direcció

```
public enum Direccio {Quiet, Amunt, Dreta, Avall, Esquerra}
```

Classe Escenari

```
public class Escenari
{
    /// <summary>
    /// Crea un escenari donades unes mides
    /// </summary>
    /// <param name="files">Número de files de l'escenari</param>
    /// <param name="columnes">Número de columnes de l'escenari</param>
    public Escenari(int files, int columnes)

    /// <summary>
    /// Retorna el número de files de l'escenari
    /// </summary>
    public int Files
    {get;}

    /// <summary>
    /// Retorna el número de columnes de l'escenari
    /// </summary>
    public int Columnes
    {get ;}

    /// <summary>
    /// Retorna el número de homes que hi ha dins de l'escenari
    /// </summary>
    public int Homes
    {get ;}

    /// <summary>
    /// Retorna el número de dones que hi ha dins de l'escenari
    /// </summary>
    public int Dones
    {get ;}

    /// <summary>
    /// Retorna el número de Cambrers que hi ha dins de l'escenari
    /// </summary>
    public int Cambrers
    {get ;}

    /// <summary>
    /// Mou una persona de (filOrig, colOrig) fins a la posicio adjacent (filDesti,c
    olDesti)
    /// Es suposa que les coordenades són vàlides, que hi ha una persona a l'origen
    i que
    /// el destí està buit.
    /// </summary>
    /// <param name="filOrig">Fila de la coordenada d'origen</param>
    /// <param name="colOrig">Columna de la coordenada d'origen</param>
    /// <param name="filDesti">Fila de la coordenada de destí</param>
    /// <param name="colDesti">Columna de la coordenada de destí</param>
    private void Moure (int filOrig, int colOrig, int filDesti, int colDesti)
```

```

/// <summary>
/// Retorna la Posició que hi ha en una coordenada donada
/// </summary>
public Posicio this [int fila, int col]
{
    get;
}

/// <summary>
/// Mira si una coordenada es correcte per ser destí d'una persona
/// </summary>
/// <param name="fil">fila de la coordenada</param>
/// <param name="col">columna de la coordenada</param>
/// <returns>retorna si la coordenada és vàlida i està buida</returns>
public bool DestiValid (int fil, int col)

/// <summary>
/// Retorna el contingut del escenari en forma de matriu d'strings,
/// representant cada persona amb el seu nom
/// </summary>
/// <returns>Matriu de caràcters</returns>
public String [,] ContingutNoms ()

/// <summary>
/// Elimina una persona de l'escenari i de la taula de persones
/// </summary>
/// <param name="fil">Fila on està la persona</param>
/// <param name="col">Columna on està la persona</param>
public void buidar (int fil, int col)

/// <summary>
/// Posa una Persona dins de l'escenari i a la taula de persones
/// Si la posició de la persona ja està ocupada, genera una excepció
/// </summary>
/// <param name="pers">Persona a afegir</param>
public void posar (Persona pers)

/// <summary>
/// Mira si en el tauler hi ha alguna persona amb aquest nom
/// </summary>
/// <param name="nom">Nom a cercar</param>
/// <returns>Si hi ha coincidència</returns>
public bool NomRepetit (string nom)

/// <summary>
/// Fa que totes les persones facin un moviment
/// </summary>
public void Cicle ()
}

```

Classe TaulaPersones

```
public class TaulaPersones
/// <summary>
/// Crea una taula de referències a persones
/// </summary>
public TaulaPersones()
/// <summary>
/// Assigna o obté una persona de la taula
/// </summary>
public Persona this [string nom]
{get; set;}

/// <summary>
/// Obtè el número total de persones
/// </summary>
public int NumPersones
{get; }

/// <summary>
/// Afegeix una persona a la taula
/// </summary>
/// <param name="conv">Convidat a afegir</param>
public void Afegir (Persona pers)

/// <summary>
/// Eliminar una persona de la taula
/// </summary>
/// <param name="conv">Convidat a eliminar</param>
public void Eliminar (Persona pers)

/// <summary>
/// Elimina la persona donat el seu nom
/// </summary>
/// <param name="posicio">Posició a eliminar</param>
public void Eliminar (string nom)
}
```