

Disseny i implementació de la base de dades d'un sistema de descàrrega d'aplicacions per a mòbils intel·ligents

ANNEXOS

Blanca Benítez Ponsa

Estudiant d'Enginyeria Tècnica en Informàtica de Gestió

Àlex Caminals Sánchez de la Campa

Consultor

14 de desembre de 2013

ÍNDEX

ANNEXOS, *pàgina 3*

ANNEX 1. BASE DE DADES, *pàgina 3*

ANNEX 2. INSERTS, *pàgina 15*

ANNEX 3. PROCEDIMENTS, *pàgina 43*

ANNEX 4. ALTRES INSERTS, *pàgina 73*

ANNEX 5. PROVES, *pàgina 77*

ANNEXOS

ANNEX 1. BASE DE DADES

```
-- SQL Manager 2011 for SQL Server 3.7.0.1
-- -----
-- Host      : 172.16.13.2,2301
-- Database  : bbenitezp
-- Version   : Microsoft SQL Server 9.00.2047.00

--
-- Definition for table aplicacions :
--

CREATE TABLE [dbo].[aplicacions] (
  [aplicacio] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
  [nom] varchar(100) COLLATE Modern_Spanish_CI_AS NULL
)
ON [PRIMARY]
GO

--
-- Definition for table paisos :
--

CREATE TABLE [dbo].[paisos] (
  [pais] varchar(2) COLLATE Modern_Spanish_CI_AS NOT NULL,
  [nom] varchar(100) COLLATE Modern_Spanish_CI_AS NULL
)
ON [PRIMARY]
GO

--
-- Definition for table dispositius_fabricants :
--

CREATE TABLE [dbo].[dispositius_fabricants] (
  [fabricant] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
  [nom] varchar(100) COLLATE Modern_Spanish_CI_AS NULL
)
ON [PRIMARY]
GO

--
-- Definition for table dispositius_models :
--

CREATE TABLE [dbo].[dispositius_models] (
  [model] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
  [nom] varchar(100) COLLATE Modern_Spanish_CI_AS NULL,
  [fabricant] varchar(10) COLLATE Modern_Spanish_CI_AS NULL
)
ON [PRIMARY]
GO
```

```
--
-- Definition for table dispositius :
--

CREATE TABLE [dbo].[dispositius] (
    [dispositiu] int NOT NULL,
    [IMEI] varchar(17) COLLATE Modern_Spanish_CI_AS NULL,
    [w] int NULL,
    [h] int NULL,
    [model] varchar(10) COLLATE Modern_Spanish_CI_AS NULL,
    [pais] varchar(2) COLLATE Modern_Spanish_CI_AS NULL
)
ON [PRIMARY]
GO

--
-- Definition for table so :
--

CREATE TABLE [dbo].[so] (
    [so] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [nom] varchar(100) COLLATE Modern_Spanish_CI_AS NULL
)
ON [PRIMARY]
GO

--
-- Definition for table versions :
--

CREATE TABLE [dbo].[versions] (
    [aplicacio] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [so] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [versio] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [data_pujada] datetime NULL,
    [binari] varchar(100) COLLATE Modern_Spanish_CI_AS NULL,
    [mida] decimal(10, 2) DEFAULT 0.00 NOT NULL,
    [wminim] int DEFAULT 0 NOT NULL,
    [hminim] int DEFAULT 0 NOT NULL,
    [actiu] int DEFAULT 0 NOT NULL
)
ON [PRIMARY]
GO

--
-- Definition for table operadors_empreses :
--

CREATE TABLE [dbo].[operadors_empreses] (
    [empresa] int NOT NULL,
    [nom] varchar(100) COLLATE Modern_Spanish_CI_AS NULL
)
ON [PRIMARY]
GO

--
```

-- Definition for table operadors :

--

```
CREATE TABLE [dbo].[operadors] (  
    [operador] int NOT NULL,  
    [nom] varchar(100) COLLATE Modern_Spanish_CI_AS NULL,  
    [empresa] int NOT NULL  
)  
ON [PRIMARY]  
GO
```

--

-- Definition for table usuaris :

--

```
CREATE TABLE [dbo].[usuaris] (  
    [usuari] int NOT NULL,  
    [nom] varchar(100) COLLATE Modern_Spanish_CI_AS NULL,  
    [cognoms] varchar(100) COLLATE Modern_Spanish_CI_AS NULL,  
    [mobil] varchar(20) COLLATE Modern_Spanish_CI_AS NULL,  
    [operador] int NULL,  
    [email] varchar(100) COLLATE Modern_Spanish_CI_AS NULL,  
    [data_baixa] datetime NULL  
)  
ON [PRIMARY]  
GO
```

--

-- Definition for table modes_pagament :

--

```
CREATE TABLE [dbo].[modes_pagament] (  
    [mode] int NOT NULL,  
    [nom] varchar(100) COLLATE Modern_Spanish_CI_AS NULL  
)  
ON [PRIMARY]  
GO
```

--

-- Definition for table descarregues :

--

```
CREATE TABLE [dbo].[descarregues] (  
    [descarrega] int IDENTITY(1, 1) NOT NULL,  
    [data] datetime NULL,  
    [aplicacio] varchar(10) COLLATE Modern_Spanish_CI_AS NULL,  
    [so] varchar(10) COLLATE Modern_Spanish_CI_AS NULL,  
    [versio] varchar(10) COLLATE Modern_Spanish_CI_AS NULL,  
    [usuari] int NULL,  
    [operador] int NULL,  
    [pais] varchar(2) COLLATE Modern_Spanish_CI_AS NULL,  
    [dispositiu] int NULL,  
    [IMEI] varchar(17) COLLATE Modern_Spanish_CI_AS NULL,  
    [preu] decimal(10, 2) DEFAULT 0.00 NOT NULL,  
    [mode] int NULL,  
    [wifi] int DEFAULT 0 NOT NULL
```

```
)
ON [PRIMARY]
GO

--
-- Definition for table descripcions :
--

CREATE TABLE [dbo].[descripcions] (
  [aplicacio] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
  [idioma] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
  [descripcio] text COLLATE Modern_Spanish_CI_AS NULL
)
ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO

--
-- Definition for table desenvolupadors_empreses :
--

CREATE TABLE [dbo].[desenvolupadors_empreses] (
  [empresa] int NOT NULL,
  [nom] varchar(100) COLLATE Modern_Spanish_CI_AS NULL
)
ON [PRIMARY]
GO

--
-- Definition for table desenvolupadors :
--

CREATE TABLE [dbo].[desenvolupadors] (
  [desenvolupador] int NOT NULL,
  [nom] varchar(100) COLLATE Modern_Spanish_CI_AS NULL,
  [empresa] int NULL,
  [representant] varchar(100) COLLATE Modern_Spanish_CI_AS DEFAULT 'PARTICULAR' NOT
NULL,
  [adreca] varchar(255) COLLATE Modern_Spanish_CI_AS NULL,
  [telefon] varchar(100) COLLATE Modern_Spanish_CI_AS NULL,
  [email] varchar(100) COLLATE Modern_Spanish_CI_AS NULL,
  [pais] varchar(2) COLLATE Modern_Spanish_CI_AS NULL
)
ON [PRIMARY]
GO

--
-- Definition for table estadistiques :
--

CREATE TABLE [dbo].[estadistiques] (
  [estadistica] int NOT NULL,
  [valor] varchar(100) COLLATE Modern_Spanish_CI_AS DEFAULT " NOT NULL,
  [param] varchar(100) COLLATE Modern_Spanish_CI_AS DEFAULT " NOT NULL,
  [data] datetime NULL
)
```

```
ON [PRIMARY]
GO
```

```
--
-- Definition for table logs :
--
```

```
CREATE TABLE [dbo].[logs] (
    [log] int IDENTITY(1, 1) NOT NULL,
    [procediment] varchar(100) COLLATE Modern_Spanish_CI_AS NULL,
    [usuari_bd] varchar(30) COLLATE Modern_Spanish_CI_AS NULL,
    [instant] datetime NULL,
    [entrada] varchar(1000) COLLATE Modern_Spanish_CI_AS NULL,
    [sortida] varchar(1000) COLLATE Modern_Spanish_CI_AS NULL,
    [rsp] varchar(100) COLLATE Modern_Spanish_CI_AS NULL
)
ON [PRIMARY]
GO
```

```
--
-- Definition for table versions_desenvolupadors :
--
```

```
CREATE TABLE [dbo].[versions_desenvolupadors] (
    [aplicacio] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [so] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [versio] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [desenvolupador] int NOT NULL
)
ON [PRIMARY]
GO
```

```
--
-- Definition for table versions_preus :
--
```

```
CREATE TABLE [dbo].[versions_preus] (
    [aplicacio] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [so] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [versio] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [pais] varchar(2) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [preu] decimal(10, 2) DEFAULT 0.00 NOT NULL
)
ON [PRIMARY]
GO
```

```
--
-- Definition for table versions_url :
--
```

```
CREATE TABLE [dbo].[versions_url] (
    [aplicacio] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [so] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [versio] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [idioma] varchar(10) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [url] varchar(512) COLLATE Modern_Spanish_CI_AS NULL
)
```

```
)
ON [PRIMARY]
GO

--
-- Definition for indices :
--

ALTER TABLE [dbo].[aplicacions]
ADD PRIMARY KEY CLUSTERED ([aplicacio])
WITH (
    PAD_INDEX = OFF,
    IGNORE_DUP_KEY = OFF,
    STATISTICS_NORECOMPUTE = OFF,
    ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
GO

ALTER TABLE [dbo].[paisos]
ADD PRIMARY KEY CLUSTERED ([pais])
WITH (
    PAD_INDEX = OFF,
    IGNORE_DUP_KEY = OFF,
    STATISTICS_NORECOMPUTE = OFF,
    ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
GO

ALTER TABLE [dbo].[dispositius_fabricants]
ADD PRIMARY KEY CLUSTERED ([fabricant])
WITH (
    PAD_INDEX = OFF,
    IGNORE_DUP_KEY = OFF,
    STATISTICS_NORECOMPUTE = OFF,
    ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
GO

ALTER TABLE [dbo].[dispositius_models]
ADD PRIMARY KEY CLUSTERED ([model])
WITH (
    PAD_INDEX = OFF,
    IGNORE_DUP_KEY = OFF,
    STATISTICS_NORECOMPUTE = OFF,
    ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
GO

CREATE UNIQUE NONCLUSTERED INDEX [dispositius_uq] ON [dbo].[dispositius]
([IMEI])
WITH (
    PAD_INDEX = OFF,
```



```
IGNORE_DUP_KEY = OFF,  
DROP_EXISTING = OFF,  
STATISTICS_NORECOMPUTE = OFF,  
SORT_IN_TEMPDB = OFF,  
ONLINE = OFF,  
ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[dispositius]  
ADD PRIMARY KEY CLUSTERED ([dispositiu])  
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[so]  
ADD PRIMARY KEY CLUSTERED ([so])  
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[versions]  
ADD PRIMARY KEY CLUSTERED ([aplicacio], [so], [versio])  
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[operadors_empreses]  
ADD PRIMARY KEY CLUSTERED ([empresa])  
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[operadors]  
ADD PRIMARY KEY CLUSTERED ([operador])
```

```
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[usuaris]  
ADD PRIMARY KEY CLUSTERED ([usuari])  
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[modes_pagament]  
ADD PRIMARY KEY CLUSTERED ([mode])  
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[descarregues]  
ADD PRIMARY KEY CLUSTERED ([descarrega])  
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[descripcions]  
ADD PRIMARY KEY CLUSTERED ([aplicacio], [idioma])  
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[desenvolupadors_empreses]  
ADD PRIMARY KEY CLUSTERED ([empresa])  
WITH (  

```

```
PAD_INDEX = OFF,  
IGNORE_DUP_KEY = OFF,  
STATISTICS_NORECOMPUTE = OFF,  
ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[desenvolupadors]  
ADD PRIMARY KEY CLUSTERED ([desenvolupador])  
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[estadistiques]  
ADD PRIMARY KEY CLUSTERED ([estadistica], [param])  
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[logs]  
ADD PRIMARY KEY CLUSTERED ([log])  
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[versions_desenvolupadors]  
ADD PRIMARY KEY CLUSTERED ([aplicacio], [so], [versio], [desenvolupador])  
WITH (  
    PAD_INDEX = OFF,  
    IGNORE_DUP_KEY = OFF,  
    STATISTICS_NORECOMPUTE = OFF,  
    ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON)  
ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[versions_preus]  
ADD CONSTRAINT [preus_idx]  
PRIMARY KEY CLUSTERED ([aplicacio], [so], [versio], [pais])  
WITH (  

```

```
PAD_INDEX = OFF,
IGNORE_DUP_KEY = OFF,
STATISTICS_NORECOMPUTE = OFF,
ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
GO

ALTER TABLE [dbo].[versions_url]
ADD PRIMARY KEY CLUSTERED ([aplicacio], [so], [versio], [idioma])
WITH (
    PAD_INDEX = OFF,
    IGNORE_DUP_KEY = OFF,
    STATISTICS_NORECOMPUTE = OFF,
    ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
GO

--
-- Definition for foreign keys :
--

ALTER TABLE [dbo].[dispositius_models]
ADD CONSTRAINT [dispositius_models_fk] FOREIGN KEY ([fabricant])
    REFERENCES [dbo].[dispositius_fabricants] ([fabricant])
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
GO

ALTER TABLE [dbo].[dispositius]
ADD CONSTRAINT [dispositius_fk] FOREIGN KEY ([model])
    REFERENCES [dbo].[dispositius_models] ([model])
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
GO

ALTER TABLE [dbo].[dispositius]
ADD CONSTRAINT [dispositius_fk2] FOREIGN KEY ([pais])
    REFERENCES [dbo].[paisos] ([pais])
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
GO

ALTER TABLE [dbo].[versions]
ADD CONSTRAINT [versions_fk] FOREIGN KEY ([aplicacio])
    REFERENCES [dbo].[aplicacions] ([aplicacio])
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
GO

ALTER TABLE [dbo].[versions]
ADD CONSTRAINT [versions_fk2] FOREIGN KEY ([so])
    REFERENCES [dbo].[so] ([so])
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
```

GO

```
ALTER TABLE [dbo].[operadors]
ADD CONSTRAINT [operadors_fk] FOREIGN KEY ([empresa])
REFERENCES [dbo].[operadors_empreses] ([empresa])
ON UPDATE NO ACTION
ON DELETE NO ACTION
```

GO

```
ALTER TABLE [dbo].[usuaris]
ADD CONSTRAINT [usuaris_fk] FOREIGN KEY ([operador])
REFERENCES [dbo].[operadors] ([operador])
ON UPDATE NO ACTION
ON DELETE NO ACTION
```

GO

```
ALTER TABLE [dbo].[descarregues]
ADD CONSTRAINT [descarregues_fk] FOREIGN KEY ([aplicacio], [so], [versio])
REFERENCES [dbo].[versions] ([aplicacio], [so], [versio])
ON UPDATE NO ACTION
ON DELETE NO ACTION
```

GO

```
ALTER TABLE [dbo].[descarregues]
ADD CONSTRAINT [descarregues_fk2] FOREIGN KEY ([usuari])
REFERENCES [dbo].[usuaris] ([usuari])
ON UPDATE NO ACTION
ON DELETE NO ACTION
```

GO

```
ALTER TABLE [dbo].[descarregues]
ADD CONSTRAINT [descarregues_fk3] FOREIGN KEY ([operador])
REFERENCES [dbo].[operadors] ([operador])
ON UPDATE NO ACTION
ON DELETE NO ACTION
```

GO

```
ALTER TABLE [dbo].[descarregues]
ADD CONSTRAINT [descarregues_fk4] FOREIGN KEY ([pais])
REFERENCES [dbo].[paisos] ([pais])
ON UPDATE NO ACTION
ON DELETE NO ACTION
```

GO

```
ALTER TABLE [dbo].[descarregues]
ADD CONSTRAINT [descarregues_fk5] FOREIGN KEY ([dispositiu])
REFERENCES [dbo].[dispositius] ([dispositiu])
ON UPDATE NO ACTION
ON DELETE NO ACTION
```

GO

```
ALTER TABLE [dbo].[descarregues]
ADD CONSTRAINT [descarregues_fk6] FOREIGN KEY ([mode])
REFERENCES [dbo].[modes_pagament] ([mode])
ON UPDATE NO ACTION
ON DELETE NO ACTION
```

GO

```
ALTER TABLE [dbo].[descripcions]
ADD CONSTRAINT [descripcions_fk] FOREIGN KEY ([aplicacio])
REFERENCES [dbo].[aplicacions] ([aplicacio])
ON UPDATE NO ACTION
ON DELETE NO ACTION
GO
```

```
ALTER TABLE [dbo].[desenvolupadors]
ADD CONSTRAINT [desenvolupadors_fk] FOREIGN KEY ([pais])
REFERENCES [dbo].[paisos] ([pais])
ON UPDATE NO ACTION
ON DELETE NO ACTION
GO
```

```
ALTER TABLE [dbo].[desenvolupadors]
ADD CONSTRAINT [desenvolupadors_fk2] FOREIGN KEY ([empresa])
REFERENCES [dbo].[desenvolupadors_empreses] ([empresa])
ON UPDATE NO ACTION
ON DELETE NO ACTION
GO
```

```
ALTER TABLE [dbo].[versions_desenvolupadors]
ADD CONSTRAINT [versions_desenvolupadors_fk] FOREIGN KEY ([aplicacio], [so], [versio])
REFERENCES [dbo].[versions] ([aplicacio], [so], [versio])
ON UPDATE NO ACTION
ON DELETE NO ACTION
GO
```

```
ALTER TABLE [dbo].[versions_desenvolupadors]
ADD CONSTRAINT [versions_desenvolupadors_fk2] FOREIGN KEY ([desenvolupador])
REFERENCES [dbo].[desenvolupadors] ([desenvolupador])
ON UPDATE NO ACTION
ON DELETE NO ACTION
GO
```

```
ALTER TABLE [dbo].[versions_preus]
ADD CONSTRAINT [versions_preus_fk] FOREIGN KEY ([aplicacio], [so], [versio])
REFERENCES [dbo].[versions] ([aplicacio], [so], [versio])
ON UPDATE NO ACTION
ON DELETE NO ACTION
GO
```

```
ALTER TABLE [dbo].[versions_preus]
ADD CONSTRAINT [versions_preus_fk2] FOREIGN KEY ([pais])
REFERENCES [dbo].[paisos] ([pais])
ON UPDATE NO ACTION
ON DELETE NO ACTION
GO
```

```
ALTER TABLE [dbo].[versions_url]
ADD CONSTRAINT [versions_url_fk] FOREIGN KEY ([aplicacio], [so], [versio])
REFERENCES [dbo].[versions] ([aplicacio], [so], [versio])
ON UPDATE NO ACTION
ON DELETE NO ACTION
```

GO

ANNEX 2. INSERTS

```
--  
-- Data for table dbo.dispositius_fabricants (LIMIT 0,500)  
--  
  
INSERT INTO [dbo].[dispositius_fabricants] ([fabricant], [nom])  
VALUES  
    (N'001', N'NOKIA')  
GO  
  
INSERT INTO [dbo].[dispositius_fabricants] ([fabricant], [nom])  
VALUES  
    (N'002', N'SAMSUNG')  
GO  
  
INSERT INTO [dbo].[dispositius_fabricants] ([fabricant], [nom])  
VALUES  
    (N'003', N'LG')  
GO  
  
INSERT INTO [dbo].[dispositius_fabricants] ([fabricant], [nom])  
VALUES  
    (N'004', N'SONY')  
GO  
  
INSERT INTO [dbo].[dispositius_fabricants] ([fabricant], [nom])  
VALUES  
    (N'005', N'APPLE')  
GO  
  
INSERT INTO [dbo].[dispositius_fabricants] ([fabricant], [nom])  
VALUES  
    (N'006', N'HUAWEI')  
GO  
  
INSERT INTO [dbo].[dispositius_fabricants] ([fabricant], [nom])  
VALUES  
    (N'007', N'ZTE')  
GO  
  
INSERT INTO [dbo].[dispositius_fabricants] ([fabricant], [nom])  
VALUES  
    (N'008', N'HTC')  
GO  
  
INSERT INTO [dbo].[dispositius_fabricants] ([fabricant], [nom])  
VALUES  
    (N'009', N'BLACKBERRY')  
GO  
  
--  
-- Data for table dbo.dispositius_models (LIMIT 0,500)
```

--

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'001001', N'LUMIA 710', N'001')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'001002', N'C6-01', N'001')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'001003', N'C2-05', N'001')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'001004', N'C2-02', N'001')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'001005', N'7230', N'001')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'001006', N'C7-00', N'001')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'002001', N'GALAXY SIII', N'002')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'002002', N'GALAXY SII', N'002')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'002003', N'GALAXY ACE II', N'002')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'002004', N'GALAXY ACE', N'002')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'003001', N'L3 E400', N'003')  
GO
```



```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'003002', N'OPTIMUS HUB E510', N'003')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'004001', N'XPERIA U', N'004')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'004002', N'XPERIA S', N'004')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'005001', N'IPHONE 4S', N'005')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'005002', N'IPHONE 5', N'005')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'006001', N'ASCEND P1', N'006')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'007001', N'SKATE', N'007')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'007002', N'SKATE PRO', N'007')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'008001', N'WILDFIRE S', N'008')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'008002', N'ONE X', N'008')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])  
VALUES  
(N'009001', N'CURVE 3G 9300', N'009')  
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])
VALUES
(N'009002', N'CURVE 9220', N'009')
GO
```

```
INSERT INTO [dbo].[dispositius_models] ([model], [nom], [fabricant])
VALUES
(N'009003', N'CURVE 9380', N'009')
GO
```

```
--
-- Data for table dbo.modes_pagament (LIMIT 0,500)
--
```

```
INSERT INTO [dbo].[modes_pagament] ([mode], [nom])
VALUES
(1, N'PAYPAL')
GO
```

```
INSERT INTO [dbo].[modes_pagament] ([mode], [nom])
VALUES
(2, N'TARGETA DE CRÈDIT')
GO
```

```
INSERT INTO [dbo].[modes_pagament] ([mode], [nom])
VALUES
(3, N'TARGETA DE DÉBIT')
GO
```

```
--
-- Data for table dbo.operadors_empreses (LIMIT 0,500)
--
```

```
INSERT INTO [dbo].[operadors_empreses] ([empresa], [nom])
VALUES
(1, N'MOVISTAR')
GO
```

```
INSERT INTO [dbo].[operadors_empreses] ([empresa], [nom])
VALUES
(2, N'VODAFONE')
GO
```

```
INSERT INTO [dbo].[operadors_empreses] ([empresa], [nom])
VALUES
(3, N'YOIGO')
GO
```

```
INSERT INTO [dbo].[operadors_empreses] ([empresa], [nom])
VALUES
(4, N'FRANCE TELECOM')
GO
```

```
--
-- Data for table dbo.operadors (LIMIT 0,500)
--
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])  
VALUES  
    (1, N'MOVISTAR ESPAÑA', 1)  
GO
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])  
VALUES  
    (2, N'MOVISTAR MÉXICO', 1)  
GO
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])  
VALUES  
    (3, N'MOVISTAR CHILE', 1)  
GO
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])  
VALUES  
    (4, N'VODAFONE FIJI', 2)  
GO
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])  
VALUES  
    (5, N'CYTAMOBILE-VODAFONE', 2)  
GO
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])  
VALUES  
    (6, N'VODAFONE ESPAÑA', 2)  
GO
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])  
VALUES  
    (7, N'VODAFONE FAROE ISLANDS', 2)  
GO
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])  
VALUES  
    (8, N'VODAFONE ICELAND', 2)  
GO
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])  
VALUES  
    (9, N'YOIGO', 3)  
GO
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])  
VALUES  
    (10, N'ORANGE ESPAÑA', 4)  
GO
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])  
VALUES  
    (11, N'ORANGE UK', 4)  
GO
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])
VALUES
(12, N'OPTIMUS', 4)
GO
```

```
INSERT INTO [dbo].[operadors] ([operador], [nom], [empresa])
VALUES
(13, N'ORANGE WITAJ', 4)
GO
```

```
--
-- Data for table dbo.paisos (LIMIT 0,500)
--
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'AD', N'ANDORRA')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'AE', N'UNITED ARAB EMIRATES')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'AF', N'AFGHANISTA')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'AG', N'ANTIGUA AND BARBUDA')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'AI', N'ANGUILLA')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'AL', N'ALBANIA')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'AM', N'ARMENIA')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'AO', N'ANGOLA')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
```

```
(N'AQ', N'ANTARCTICA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'AR', N'ARGENTINA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'AS', N'AMERICAN SAMOA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'AT', N'AUSTRIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'AU', N'AUSTRALIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'AW', N'ARUBA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'AX', N'ÅLAND ISLANDS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'AZ', N'AZERBAIJA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'B', N'BRUNEI DARUSSALAM')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BA', N'BOSNIA AND HERZEGOVINA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BB', N'BARBADOS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BD', N'BANGLADESH')
```

GO

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BE', N'BELGIUM')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BF', N'BURKINA FASO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BG', N'BULGARIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BH', N'BAHRAI')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BI', N'BURUNDI')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BJ', N'BENI')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BL', N'SAINT BARTHÉLEMY')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BM', N'BERMUDA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BO', N'BOLIVIA, PLURINATIONAL STATE OF')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BQ', N'BONAIRE, SINT EUSTATIUS AND SABA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BR', N'BRAZIL')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BS', N'BAHAMAS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BT', N'BHUTA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BV', N'BOUVET ISLAND')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BW', N'BOTSWANA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BY', N'BELARUS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'BZ', N'BELIZE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'C', N'CHINA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CA', N'CANADA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CC', N'COCOS (KEELING) ISLANDS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CD', N'CONGO, THE DEMOCRATIC REPUBLIC OF THE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CF', N'CENTRAL AFRICAN REPUBLIC')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CG', N'CONGO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CH', N'SWITZERLAND')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CI', N'CÔTE D'IVOIRE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CK', N'COOK ISLANDS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CL', N'CHILE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CM', N'CAMEROO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CO', N'COLOMBIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CR', N'COSTA RICA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CU', N'CUBA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CV', N'CAPE VERDE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CW', N'CURAÇAO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
```



```
VALUES  
(N'CX', N'CHRISTMAS ISLAND')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CY', N'CYPRUS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'CZ', N'CZECH REPUBLIC')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'DE', N'GERMANY')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'DJ', N'DJIBOUTI')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'DK', N'DENMARK')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'DM', N'DOMINICA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'DO', N'DOMINICAN REPUBLIC')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'DZ', N'ALGERIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'EC', N'ECUADOR')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'EE', N'ESTONIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES
```

```
(N'EG', N'EGYPT')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'EH', N'WESTERN SAHARA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'ER', N'ERITREA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'ES', N'SPAI')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'ET', N'ETHIOPIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'FI', N'FINLAND')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'FJ', N'FIJI')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'FK', N'FALKLAND ISLANDS (MALVINAS)')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'FM', N'MICRONESIA, FEDERATED STATES OF')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'FO', N'FAROE ISLANDS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'FR', N'FRANCE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'G', N'GUINEA')
```

GO

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GA', N'GABO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GB', N'UNITED KINGDOM')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GD', N'GRENADA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GE', N'GEORGIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GF', N'FRENCH GUIANA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GG', N'GUERNSEY')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GH', N'GHANA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GI', N'GIBRALTAR')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GL', N'GREENLAND')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GM', N'GAMBIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GP', N'GUADELOUPE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GQ', N'EQUATORIAL GUINEA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GR', N'GREECE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GS', N'SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GT', N'GUATEMALA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GU', N'GUAM')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GW', N'GUINEA-BISSAU')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'GY', N'GUYANA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'H', N'HONDURAS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'HK', N'HONG KONG')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'HM', N'HEARD ISLAND AND MCDONALD ISLANDS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'HR', N'CROATIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'HT', N'HAITI')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'HU', N'HUNGARY')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'IN', N'INDIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'ID', N'INDONESIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'IE', N'IRELAND')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'IL', N'ISRAEL')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'IM', N'ISLE OF MA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'IO', N'BRITISH INDIAN OCEAN TERRITORY')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'IQ', N'IRAQ')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'IR', N'IRAN, ISLAMIC REPUBLIC OF')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'IS', N'ICELAND')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
```

```
VALUES  
(N'IT', N'ITALY')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'JE', N'JERSEY')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'JM', N'JAMAICA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'JO', N'JORDA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'JP', N'JAPA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'K', N'SAINT KITTS AND NEVIS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'KE', N'KENYA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'KG', N'KYRGYZSTA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'KH', N'CAMBODIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'KI', N'KIRIBATI')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'KM', N'COMOROS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES
```

```
(N'KP', N'KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'KR', N'KOREA, REPUBLIC OF')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'KW', N'KUWAIT')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'KY', N'CAYMAN ISLANDS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'KZ', N'KAZAKHSTA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'LA', N'LAO PEOPLE'S DEMOCRATIC REPUBLIC')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'LB', N'LEBANO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'LC', N'SAINT LUCIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'LI', N'LIECHTENSTEI')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'LK', N'SRI LANKA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'LR', N'LIBERIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'LS', N'LESOTHO')
```

GO

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'LT', N'LITHUANIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'LU', N'LUXEMBOURG')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'LV', N'LATVIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'LY', N'LIBYAN ARAB JAMAHIRIYA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'M', N'MONGOLIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MA', N'MOROCCO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MC', N'MONACO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MD', N'MOLDOVA, REPUBLIC OF')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'ME', N'MONTENEGRO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MF', N'SAINT MARTIN (FRENCH PART)')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MG', N'MADAGASCAR')  
GO
```



```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MH', N'MARSHALL ISLANDS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MK', N'MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'ML', N'MALI')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MM', N'MYANMAR')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MO', N'MACAO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MP', N'NORTHERN MARIANA ISLANDS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MQ', N'MARTINIQUE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MR', N'MAURITANIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MS', N'MONTSERRAT')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MT', N'MALTA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'MU', N'MAURITIUS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'MV', N'MALDIVES')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'MW', N'MALAWI')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'MX', N'MEXICO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'MY', N'MALAYSIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'MZ', N'MOZAMBIQUE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'NA', N'NAMIBIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'NC', N'NEW CALEDONIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'NE', N'NIGER')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'NF', N'NORFOLK ISLAND')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'NG', N'NIGERIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
    (N'NI', N'NICARAGUA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
```

```
VALUES
(N'NL', N'NETHERLANDS[NOTE 1]')
GO

INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'NO', N'NORWAY')
GO

INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'NP', N'NEPAL')
GO

INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'NR', N'NAURU')
GO

INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'NU', N'NIUE')
GO

INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'NZ', N'NEW ZEALAND')
GO

INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'OM', N'OMA')
GO

INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'P', N'PITCAIR')
GO

INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'PA', N'PANAMA')
GO

INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'PE', N'PERU')
GO

INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'PF', N'FRENCH POLYNESIA')
GO

INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
```

```
(N'PG', N'PAPUA NEW GUINEA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'PH', N'PHILIPPINES')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'PK', N'PAKISTA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'PL', N'POLAND')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'PM', N'SAINT PIERRE AND MIQUELO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'PR', N'PUERTO RICO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'PS', N'PALESTINIAN TERRITORY, OCCUPIED')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'PT', N'PORTUGAL')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'PW', N'PALAU')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'PY', N'PARAGUAY')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'QA', N'QATAR')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'RE', N'RÉUNIO')
```

GO

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'RO', N'ROMANIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'RS', N'SERBIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'RU', N'RUSSIAN FEDERATIO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'RW', N'RWANDA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'S', N'SENEGAL')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'SA', N'SAUDI ARABIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'SB', N'SOLOMON ISLANDS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'SC', N'SEYCHELLES')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'SD', N'SUDA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'SE', N'SWEDE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'SG', N'SINGAPORE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'SH', N'SAINT HELENA, ASCENSION AND TRISTAN DA CUNHA')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'SI', N'SLOVENIA')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'SJ', N'SVALBARD AND JAN MAYE')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'SK', N'SLOVAKIA')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'SL', N'SIERRA LEONE')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'SM', N'SAN MARINO')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'SO', N'SOMALIA')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'SR', N'SURINAME')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'SS', N'SOUTH SUDA')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'ST', N'SAO TOME AND PRINCIPE')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
VALUES
(N'SV', N'EL SALVADOR')
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'SX', N'SINT MAARTE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'SY', N'SYRIAN ARAB REPUBLIC')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'SZ', N'SWAZILAND')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'T', N'TUNISIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TC', N'TURKS AND CAICOS ISLANDS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TD', N'CHAD')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TF', N'FRENCH SOUTHERN TERRITORIES')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TG', N'TOGO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TH', N'THAILAND')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TJ', N'TAJIKISTA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TK', N'TOKELAU')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])
```

```
VALUES  
(N'TL', N'TIMOR-LESTE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TM', N'TURKMENISTA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TO', N'TONGA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TR', N'TURKEY')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TT', N'TRINIDAD AND TOBAGO')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TV', N'TUVALU')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TW', N'TAIWAN, PROVINCE OF CHINA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'TZ', N'TANZANIA, UNITED REPUBLIC OF')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'UA', N'UKRAINE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'UG', N'UGANDA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'UM', N'UNITED STATES MINOR OUTLYING ISLANDS')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES
```



```
(N'US', N'UNITED STATES')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'UY', N'URUGUAY')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'UZ', N'UZBEKISTA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'V', N'VIET NAM')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'VA', N'HOLY SEE (VATICAN CITY STATE'))  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'VC', N'SAINT VINCENT AND THE GRENADINES')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'VE', N'VENEZUELA, BOLIVARIAN REPUBLIC OF')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'VG', N'VIRGIN ISLANDS, BRITISH')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'VI', N'VIRGIN ISLANDS, U.S.')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'VU', N'VANUATU')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'WF', N'WALLIS AND FUTUNA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'WS', N'SAMOA')
```

GO

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'YE', N'YEME')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'YT', N'MAYOTTE')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'ZA', N'SOUTH AFRICA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'ZM', N'ZAMBIA')  
GO
```

```
INSERT INTO [dbo].[paisos] ([pais], [nom])  
VALUES  
(N'ZW', N'ZIMBABWE')  
GO
```

```
--  
-- Data for table dbo.so (LIMIT 0,500)  
--
```

```
INSERT INTO [dbo].[so] ([so], [nom])  
VALUES  
(N'AND', N'ANDROID')  
GO
```

```
INSERT INTO [dbo].[so] ([so], [nom])  
VALUES  
(N'BB', N'BLACKBERRY OS')  
GO
```

```
INSERT INTO [dbo].[so] ([so], [nom])  
VALUES  
(N'IOS', N'IOS')  
GO
```

```
INSERT INTO [dbo].[so] ([so], [nom])  
VALUES  
(N'LI', N'LINUX')  
GO
```

```
INSERT INTO [dbo].[so] ([so], [nom])  
VALUES  
(N'SYM', N'SYMBIAN OS')  
GO
```

```
INSERT INTO [dbo].[so] ([so], [nom])
VALUES
(N'WDP', N'WINDOWS PHONE')
GO
```

ANNEX 3. PROCEDIMENTS

```
--
-- Definition for stored procedure aplicacioAlta :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Procediment per donar d'alta una aplicació
CREATE PROCEDURE dbo.aplicacioAlta(
    @aplicacio varchar(10), @so varchar(10), @versio varchar(10),
    @binari varchar(100),
    @mida decimal(10,2),
    @wminim integer,
    @hminim integer,
    @actiu integer,
    @url varchar(512),
    @preu decimal(10,2),
    @RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    IF ((@aplicacio='') OR (@so='') OR (@versio=''))
    BEGIN
        --Comprovem que s'ha especificat el codi de l'aplicació
        SET @RSP='ERROR: No ha especificat el codi de l'aplicació a donar d'alta '
    END ELSE
    BEGIN
        --Mirem si existeixen aquest codis d'aplicació i de so
        SELECT @existeix = COUNT(*) FROM aplicacions WHERE aplicacio = @aplicacio
    IF (@existeix=0)
        BEGIN
            --Si no existeix una aplicació amb aquest codi retornem un error
            SET @RSP='ERROR: No existeix una aplicació amb aquest codi'
        END ELSE
        BEGIN
            SELECT @existeix = COUNT(*) FROM so WHERE so = @so
        IF (@existeix=0)
            BEGIN
                --Si no existeix un so amb aquest codi retornem un error
                SET @RSP='ERROR: No existeix un so amb aquest codi'
            END ELSE
            BEGIN
                SET @RSP='OK'
            END
        END
    END
END
```

```
--Mirem si existeix una aplicació amb aquest codi
SELECT @existeix = COUNT(*) FROM versions WHERE aplicacio
= @aplicacio AND so = @so AND versio = @versio
IF (@existeix>0)
BEGIN
    --Si existeix una aplicació amb aquest codi retornem un error
    SET @RSP='ERROR: Ja existeix una aplicació amb aquest codi'
END ELSE
BEGIN
    --Si no existeix una aplicació amb aquest codi la donem d'alta
    --Considerarem com a data de pujada la del moment actual
    --Per defecte l'idioma en que s'introdueix la url és l'anglès (EN)
    --Per defecte el país pel qual s'introdueix el preu és per Espanya (ES)
    BEGIN TRY
        BEGIN TRAN
            INSERT INTO versions (aplicacio, so, versio,
data_pujada, binari, mida, wminim, hminim, actiu)
            VALUES (@aplicacio, @so, @versio, GETDATE(), @binari,
@mida, @wminim, @hminim, @actiu)
            INSERT INTO versions_url (aplicacio, so, versio, idioma, url) VALUES
(@aplicacio, @so, @versio, 'EN', @url)
            INSERT INTO versions_preus (aplicacio, so, versio, pais, preu) VALUES
(@aplicacio, @so, @versio, 'ES', @preu)
        COMMIT TRAN
    END TRY
    BEGIN CATCH
        --Si hi ha algun error al fer els diferents inserts no es farà
        --cap d'ells i retornarem el missatge d'error
        ROLLBACK TRAN
        SET @RSP='ERROR: ' + ERROR_MESSAGE()
    END CATCH
END
END
END
END

--Recollim l'usuari i fer l'insert a logs
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('aplicacioAlta', @usu, GETDATE(), @aplicacio + '~' + @so + '~' + @versio + '~' +
@binari + '~' + CAST(@mida AS VARCHAR) + '~' + CAST(@wminim AS VARCHAR) + '~' +
CAST(@hminim AS VARCHAR) + '~' + CAST(@actiu AS VARCHAR), @RSP)
--No afegim ni la url ni el preu en la columna entrada
END
GO

--
-- Definition for stored procedure aplicacioBaixa :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Les aplicacions no s'eliminen mai de la base de dades, sino que es
```

```
--marquen com a no actives.
CREATE PROCEDURE dbo.aplicacioBaixa(@aplicacio varchar(10), @so varchar(10), @versio
varchar(10), @RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    IF ((@aplicacio<>'') AND (@so<>'') AND (@versio<>''))
    BEGIN
        --Mirem si existeix una aplicació amb aquest codi
        SELECT @existeix = COUNT(*) FROM versions WHERE aplicacio = @aplicacio
        AND so = @so AND versio = @versio
        IF (@existeix>0)
        BEGIN
            --Si existeix una aplicació amb aquest codi cal veure que estigui activa
            SELECT @existeix = COUNT(*) FROM versions WHERE aplicacio =
            @aplicacio AND so = @so AND versio = @versio AND actiu = 1
            IF (@existeix=0)
            BEGIN
                --L'aplicació ja no està activa
                SET @RSP='ERROR: Aquesta aplicació ja està desactivada '
            END ELSE
            BEGIN
                BEGIN TRY
                    UPDATE versions SET actiu = 0 WHERE aplicacio =
                    @aplicacio AND so = @so AND versio = @versio
                END TRY
                BEGIN CATCH
                    --Si l'update no fos satisfactori retornarem el missatge d'error
                    SET @RSP='ERROR: ' + ERROR_MESSAGE()
                END CATCH;
            END
        END ELSE
        BEGIN
            --Si no existeix una aplicació amb aquest codi retornem un error
            SET @RSP='ERROR: No existeix l'aplicació'
        END ELSE
        BEGIN
            --Comprovem que s'ha especificat el codi de l'aplicació
            SET @RSP='ERROR: No ha especificat el codi de l'aplicació a donar de baixa'
        END

        --Recollim l'usuari i fer l'insert a logs
        DECLARE @usu varchar(30)
        SELECT @usu = SYSTEM_USER
        INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
        VALUES ('aplicacioBaixa', @usu, GETDATE(), @aplicacio + '~' + @so + '~' + @versio, '', @RSP)
    END
GO

--
```

```
-- Definition for stored procedure aplicacioModificacio :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Procediment per activar una aplicació
CREATE PROCEDURE dbo.aplicacioModificacio(@aplicacio varchar(10), @so varchar(10),
@versio varchar(10), @RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    IF ((@aplicacio='') OR (@so='') OR (@versio=''))
    BEGIN
        --Comprovem que s'ha especificat el codi de l'aplicació
        SET @RSP='ERROR: No ha especificat el codi de l'aplicació a modificar'
    END ELSE
    BEGIN
        --Mirem si existeix una aplicació amb aquest codi
        SELECT @existeix = COUNT(*) FROM versions WHERE aplicacio = @aplicacio
        AND so = @so AND versio = @versio
        IF (@existeix>0)
        BEGIN
            --Mirem en quin estat està l'aplicació
            SELECT @existeix = COUNT(*) FROM versions WHERE aplicacio = @aplicacio AND so =
            @so AND versio = @versio AND actiu = 1
            IF (@existeix=0)
            BEGIN
                BEGIN TRY
                    UPDATE versions SET actiu = 1 WHERE aplicacio =
                    @aplicacio AND so = @so AND versio = @versio
                END TRY
                BEGIN CATCH
                    --Si l'update no fos satisfactori retornarem el missatge d'error
                    SET @RSP='ERROR: ' + ERROR_MESSAGE()
                END CATCH;
            END ELSE
            BEGIN
                --L'aplicació ja està activa
                SET @RSP='ERROR: Aquesta aplicació ja està activa'
            END
        END ELSE
        BEGIN
            --Si no existeix una aplicació amb aquest codi retornem un error
            SET @RSP='ERROR: No existeix una aplicació amb aquest codi'
        END
    END

    --Recollim l'usuari i fer l'insert a logs
    DECLARE @usu varchar(30)
```

```
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('aplicacioModificacio',@usu,GETDATE()),@aplicacio + '~' + @so + '~' +
@versio,',@RSP)
END
GO
```

```
--
-- Definition for stored procedure desenvolupadorAlta :
--
```

```
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO
```

```
--Procediment per donar d'alta un desenvolupador
CREATE PROCEDURE dbo.desenvolupadorAlta(
```

```
    @desenvolupador integer,
    @nom varchar(100),
    @empresa integer,
    @representant varchar(100),
    @adreca varchar(255),
    @telefon varchar(100),
    @email varchar(100),
    @pais varchar(2),
    @RSP varchar(255) OUTPUT)
```

```
AS
```

```
BEGIN
```

```
    DECLARE @existeix integer
```

```
    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0
```

```
    --Com a dades obligatòries tindrem el codi, el nom i un telèfon
    IF ((@desenvolupador='') OR (@nom='') OR (@telefon=''))
    BEGIN
```

```
        --Comprovem que s'han especificat les dades obligatòries
        SET @RSP='ERROR: Es necessari especificar el codi, el nom i un telèfon del nou
desenvolupador per donar-lo d'alta '
```

```
    END ELSE
```

```
    BEGIN
```

```
        --Mirem si existeixen aquest codi de desenvolupador
```

```
        SELECT @existeix = COUNT(*) FROM desenvolupadors WHERE desenvolupador
= @desenvolupador
```

```
        IF (@existeix>0)
```

```
        BEGIN
```

```
            --Si existeix un desenvolupador amb aquest codi retornem un error
```

```
            SET @RSP='ERROR: Ja existeix un desenvolupador amb aquest codi'
```

```
        END ELSE
```

```
    BEGIN
```

```
        SET @existeix=-1
```

```
            IF (@empresa<>")
```

```
            BEGIN
```

```
            --Mirem si existeixen aquest codi d'empresa
```

```

SELECT      @existeix      =      COUNT(*)      FROM
desenvolupadors_empreses WHERE empresa = @empresa
IF (@existeix=0)
BEGIN
    --Si no existeix una empresa amb aquest codi retornem un error
    SET @RSP='ERROR: No existeix una empresa amb aquest codi'
END

END

IF (@existeix<>0)
BEGIN
    --Mirem si existeix el codi de país
    SELECT @existeix = COUNT(*) FROM paisos WHERE pais = @pais
    IF (@existeix=0)
    BEGIN
        --Si no existeix un país amb aquest codi retornem un error
        SET @RSP='ERROR: No existeix un país amb aquest codi'
    END ELSE
    BEGIN
        BEGIN TRY
            IF (@representant='') SET @representant='PARTICULAR'
            IF (@empresa<>")
            BEGIN
                INSERT INTO desenvolupadors (desenvolupador, nom,
                empresa, representant, adreca, telefon, email, pais)
                VALUES (@desenvolupador, @nom, @empresa,
                @representant, @adreca, @telefon, @email, @pais)
            END ELSE
            BEGIN
                INSERT INTO desenvolupadors (desenvolupador, nom,
                empresa, representant, adreca, telefon, email, pais)
                VALUES (@desenvolupador, @nom, NULL,
                @representant, @adreca, @telefon, @email, @pais)
            END
        END TRY
        BEGIN CATCH
            --Si hi ha algun error al fer l'insert retornarem el missatge
            SET @RSP='ERROR: ' + ERROR_MESSAGE()
        END CATCH
    END
END
END
END

--Recollim l'usuari i fer l'insert a logs
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('desenvolupadorAlta',@usu,GETDATE(),CAST(@desenvolupador AS
VARCHAR) + '~' + @nom + '~' + CAST(@empresa AS VARCHAR) + '~' + @representant + '~' +
@adreca + '~' + @telefon + '~' + @email + '~' + @pais,',',@RSP)
END
GO

--

```



```
-- Definition for stored procedure desenvolupadorBaixa :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Per eliminar un desenvolupador és necessari que no hagi participat en cap
--de les versions (actives o no)
CREATE PROCEDURE dbo.desenvolupadorBaixa(@desenvolupador integer, @RSP varchar(255)
OUTPUT)
AS
BEGIN
    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    IF (@desenvolupador<>")
    BEGIN
        --Mirem si existeix un desenvolupador amb aquest codi
        SELECT @existeix = COUNT(*) FROM desenvolupadors WHERE desenvolupador
= @desenvolupador
        IF (@existeix>0)
        BEGIN
            --Si existeix una versió desenvolupada per aquest
            SELECT @existeix = COUNT(*) FROM versions_desenvolupadors
WHERE desenvolupador = @desenvolupador
            IF (@existeix>0)
            BEGIN
                --El desenvolupador ha participat en alguna de les versions
                SET @RSP='ERROR: Aquest desenvolupador ha participat en
alguna de les versions. No és possible eliminar-lo.'
                END ELSE
                BEGIN
                    BEGIN TRY
                        DELETE FROM desenvolupadors WHERE desenvolupador
= @desenvolupador
                    END TRY
                    BEGIN CATCH
                        --Si el delete no fos satisfactori retornarem el missatge d'error
                        SET @RSP='ERROR: ' + ERROR_MESSAGE()
                    END CATCH;
                END
            END ELSE
            BEGIN
                --Si no existeix un desenvolupador amb aquest codi retornem un error
                SET @RSP='ERROR: No existeix aquest desenvolupador'
            END ELSE
            BEGIN
                --Comprovem que s'ha especificat el codi de desenvolupador
                SET @RSP='ERROR: No ha especificat el codi del desenvolupador a donar de
baixa'
            END
        END
    END
```

```
--Recollim l'usuari i fer l'insert a logs
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('desenvolupadorBaixa', @usu, GETDATE(), @desenvolupador, @RSP)
END
GO

--
-- Definition for stored procedure desenvolupadorModificacio :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Procediment per modificar algunes de les dades d'un desenvolupador
CREATE PROCEDURE dbo.desenvolupadorModificacio(
    @desenvolupador integer,
    @nom varchar(100),
    @empresa integer,
    @representant varchar(100),
    @adreca varchar(255),
    @telefon varchar(100),
    @email varchar(100),
    @pais varchar(2),
    @RSP varchar(255) OUTPUT)
AS
BEGIN
    --Tots els camps d'un desenvolupador són modificables excepte el seu codi

    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    IF (@desenvolupador='')
    BEGIN
        --Comprovem que s'ha especificat el codi del desenvolupador
        SET @RSP='ERROR: No ha especificat el codi del desenvolupador a modificar'
    END ELSE
    BEGIN
        IF ((@nom='') AND (@empresa='') AND (@representant='') AND (@adreca='') AND
            (@telefon='') AND (@email='') AND (@pais=''))
        BEGIN
            --Comprovem que com a mínim un dels paràmetres té valor
            SET @RSP='ERROR: Cal especificar les dades del desenvolupador a
modificar'
        END ELSE
        BEGIN
            --Mirem si existeix un desenvolupador amb aquest codi
            SELECT @existeix = COUNT(*) FROM desenvolupadors WHERE
desenvolupador = @desenvolupador
            IF (@existeix>0)
```

```

BEGIN
    --Comprovem que l'empresa existeix (en el cas que vingui plena)
    SET @existeix=-1
    IF (@empresa>0)
    BEGIN
        SELECT      @existeix      =      COUNT(*)      FROM
desenvolupadors_empreses WHERE empresa = @empresa
        IF (@existeix=0)
        BEGIN
            --Si no existeix una empresa amb aquest codi retornem un
error
            SET @RSP='ERROR: No existeix una empresa amb
aquest codi'
        END
    END

    IF (@@existeix<>0)
    BEGIN
        --Mirem si existeix el codi de país (en el cas que vingui ple)
        SET @existeix=-1
        IF (@pais<>")
        BEGIN
            SELECT @existeix = COUNT(*) FROM paisos
WHERE pais = @pais
            IF (@existeix=0)
            BEGIN
                --Si no existeix un país amb aquest codi retornem
un error
                SET @RSP='ERROR: No existeix un país amb
aquest codi'
            END
        END
    END

    IF (@@existeix<>0)
    BEGIN
        BEGIN TRY
            DECLARE @UPDATE varchar(1000)
            SET @UPDATE="
                                IF      (@nom<>")      SET
@UPDATE=@UPDATE + 'nom = ' + @nom + ' '
            IF (@empresa>0)
            BEGIN
                IF (@UPDATE<>") SET @UPDATE=@UPDATE + ', '
                SET @UPDATE=@UPDATE + 'empresa = ' + CAST(@empresa AS
VARCHAR) + ' '
            END
            IF (@representant<>")
            BEGIN
                IF (@UPDATE<>") SET @UPDATE=@UPDATE + ', '
                SET @UPDATE=@UPDATE + 'representant = ' + @representant + ' '
            END
            IF (@adreca<>")
            BEGIN
                IF (@UPDATE<>") SET @UPDATE=@UPDATE + ', '
                SET @UPDATE=@UPDATE + 'adreca = ' + @adreca + ' '
            END
        END TRY
    END

```

```

        IF (@telefon<>")
        BEGIN
            IF (@UPDATE<>") SET @UPDATE=@UPDATE + ','
            SET @UPDATE=@UPDATE + 'telefon = ' + @telefon + ' '
        END
        IF (@email<>")
        BEGIN
            IF (@UPDATE<>") SET @UPDATE=@UPDATE + ','
            SET @UPDATE=@UPDATE + 'email = ' + @email + ' '
        END
        IF (@pais<>")
        BEGIN
            IF (@UPDATE<>") SET @UPDATE=@UPDATE + ','
            SET @UPDATE=@UPDATE + 'pais = ' + @pais + ' '
        END

        SET @UPDATE='UPDATE desenvolupadors SET ' + @UPDATE +
'WHERE desenvolupador = ' + CAST(@desenvolupador AS VARCHAR)
        EXEC (@UPDATE)

        END TRY
        BEGIN CATCH
            --Si l'update no fos satisfactori retornarem el
            missatge d'error
            SET @RSP='ERROR: ' +
            ERROR_MESSAGE()
        END CATCH

        END
        END
        END ELSE
        BEGIN
            --Si no existeix un desenvolupador amb aquest codi retornem un error
            SET @RSP='ERROR: No existeix un desenvolupador amb aquest codi'
        END
        END
    END

    --Recollim l'usuari i fer l'insert a logs
    DECLARE @usu varchar(30)
    SELECT @usu = SYSTEM_USER
    INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
    VALUES ('desenvolupadorModificacio', @usu, GETDATE(), CAST(@desenvolupador AS
VARCHAR) + '~' + @nom + '~' + CAST(@empresa AS VARCHAR) + '~' + @representant + '~' +
@adreca + '~' + @telefon + '~' + @email + '~' + @pais, @RSP)
    END
    GO

    --
    -- Definition for stored procedure usuariosAlta :
    --
    GO
    SET ANSI_NULLS ON
    SET QUOTED_IDENTIFIER ON
    GO

    --Procediment per donar d'alta un usuari.

```

```
CREATE PROCEDURE dbo.usuarisAlta(@usuari integer, @nom varchar(100), @cognoms
varchar(100), @mobil varchar(20), @operador integer, @email varchar(100), @RSP varchar(255)
OUTPUT)
AS
BEGIN
    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    --no considerem l'email com a obligatori
    IF ((@usuari='') OR (@nom='') OR (@cognoms='') OR (@mobil='') OR (@operador=''))
    BEGIN
        --Comprovem que s'han especificat els paràmetres obligatoris
        SET @RSP='ERROR: Ha d'especificar totes les dades obligatòries de l'usuari'
    END ELSE
    BEGIN
        --Mirem si existeix un usuari amb aquest codi
        SELECT @existeix = COUNT(*) FROM usuaris WHERE usuari = @usuari
        IF (@existeix>0)
        BEGIN
            --Si existeix un usuari amb aquest codi i/o nom retornem un error
            SET @RSP='ERROR: Ja existeix un usuari amb aquest codi'
        END ELSE
        BEGIN
            SELECT @existeix = COUNT(*) FROM usuaris WHERE mobil = @mobil AND data_baixa
            IS NULL
            IF (@existeix>0)
            BEGIN
                --Si existeix un usuari amb aquest mobil i encara està actiu retornem un
                error
                SET @RSP='ERROR: Ja existeix un usuari actiu amb aquest número de
                mòbil'
            END ELSE
            BEGIN
                SELECT @existeix = COUNT(*) FROM operadors WHERE operador = @operador
                IF (@existeix>0)
                BEGIN
                    --Podem donar d'alta l'usuari
                    BEGIN TRY
                        INSERT INTO usuaris (usuari, nom, cognoms,
                        mobil, operador, email) VALUES (@usuari, @nom, @cognoms, @mobil, @operador, @email)
                    END TRY
                    BEGIN CATCH
                        --Si l'insert no fos satisfactori retornarem el missatge d'error
                        SET @RSP='ERROR1111: ' +
                        ERROR_MESSAGE()
                    END CATCH;
                END ELSE
                BEGIN
                    --Si no existeix un operador amb aquest codi retornem un error
                    SET @RSP='ERROR: No existeix un operador amb aquest codi'
                END
            END
        END
    END
```

```
END

--Recollim l'usuari i fer l'insert a logs
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('usuarisAlta',@usu,GETDATE(),CAST(@usuari AS VARCHAR) + '~' + @nom + '~' +
@cognoms + '~' + @mobil + '~' + CAST(@operador AS VARCHAR) + '~' + @email,',',@RSP)
END
GO

--
-- Definition for stored procedure usuarisBaixa :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Els usuaris no s'eliminen mai de la base de dades, sino que es
--marquen com a no actius introduint la data de baixa.
CREATE PROCEDURE dbo.usuarisBaixa(@usuari integer, @RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    IF (@usuari='')
    BEGIN
        --Comprovem que s'ha especificat el codi d'usuari
        SET @RSP='ERROR: No ha especificat el codi d'usuari '
    END ELSE
    BEGIN
        --Mirem si existeix un usuari amb aquest codi
        SELECT @existeix = COUNT(*) FROM usuaris WHERE usuari = @usuari
        IF (@existeix>0)
        BEGIN
            BEGIN TRY
                UPDATE usuaris SET data_baixa = GETDATE() WHERE usuari =
@usuari
            END TRY
            BEGIN CATCH
                --Si l'update no fos satisfactori retornarem el missatge d'error
                SET @RSP='ERROR: ' + ERROR_MESSAGE()
            END CATCH;
        END ELSE
        BEGIN
            --Si no existeix un usuari amb aquest codi retornem un error
            SET @RSP='ERROR: No existeix un usuari amb aquest codi'
        END
    END
END

--Recollim l'usuari i fer l'insert a logs
```

```
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('usuarisBaixa', @usu, GETDATE(), @usuari, @RSP)
END
GO

--
-- Definition for stored procedure usuarisModificacio :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Procediment per modificar algunes de les dades d'un usuari
CREATE PROCEDURE dbo.usuarisModificacio(@usuari integer, @nom varchar(100), @cognoms
varchar(100), @operador integer, @email varchar(100), @RSP varchar(255) OUTPUT)
AS
BEGIN
    --Camps modificables: nom, cognoms, operador, email
    --El número de mòbil no és modificable, si es necessari fer un canvi, cal donar de baixa
    --l'usuari i donar-ne un altre d'alta amb les dades correctes

    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    IF (@usuari='')
    BEGIN
        --Comprovem que s'ha especificat el codi de l'usuari
        SET @RSP='ERROR: No ha especificat el codi de l'usuari a modificar'
    END ELSE
    BEGIN
        IF ((@nom='') AND (@cognoms='') AND (@operador='') AND (@email=''))
        BEGIN
            --Comprovem que com a mínim un dels paràmetres té valor
            SET @RSP='ERROR: Cal especificar les dades de l'usuari a modificar'
        END ELSE
        BEGIN
            --Mirem si existeix un usuari amb aquest codi
            SELECT @existeix = COUNT(*) FROM usuaris WHERE usuari = @usuari
            IF (@existeix>0)
            BEGIN
                --Comprovem que aquest usuari està actiu
                SELECT @existeix = COUNT(*) FROM usuaris WHERE usuari =
                @usuari AND data_baixa IS NULL
                IF (@existeix>0)
                BEGIN
                    --Comprovem que l'operador existeix (en el cas que vingui
                    SET @existeix=-1
                END
            END ELSE
            IF (@operador>0)
            BEGIN
```

```

SELECT @existeix = COUNT(*) FROM operators
WHERE operador = @operador
        IF (@existeix=0)
        BEGIN
            --Si l'operador no existeix retornem un error
            SET @RSP='ERROR: Aquest codi d"operador no
existeix'
        END
    END
    IF (@existeix<>0)
    BEGIN
        BEGIN TRY
            DECLARE @UPDATE varchar(1000)
            SET @UPDATE=""
            IF (@nom<>") SET
@UPDATE=@UPDATE + 'nom = ' + @nom + ' '
            IF (@cognoms<>")
            BEGIN
                IF (@UPDATE<>") SET @UPDATE=@UPDATE + ', '
                SET @UPDATE=@UPDATE + 'cognoms = ' + @cognoms + ' '
            END
            IF (@operador<>")
            BEGIN
                IF (@UPDATE<>") SET @UPDATE=@UPDATE + ', '
                SET @UPDATE=@UPDATE + 'operador = ' + CAST(@operador AS
VARCHAR) + ' '
            END
            IF (@email<>")
            BEGIN
                IF (@UPDATE<>") SET @UPDATE=@UPDATE + ', '
                SET @UPDATE=@UPDATE + 'email = ' + @email + ' '
            END
            SET @UPDATE='UPDATE usuarios SET ' + @UPDATE + 'WHERE usuari =
' + CAST(@usuari AS VARCHAR)
            EXEC (@UPDATE)
        END TRY
        BEGIN CATCH
            --Si l'update no fos satisfactori retornarem el
missatge d'error
            SET @RSP='ERROR: ' +
ERROR_MESSAGE()
        END CATCH
    END
END ELSE
    BEGIN
        --Si l'usuari no està actiu codi retornem un error
        SET @RSP='ERROR: L"usuari no està actiu'
    END
END ELSE
    BEGIN
        --Si no existeix un usuari amb aquest codi retornem un error
        SET @RSP='ERROR: No existeix un usuari amb aquest codi'
    END
END
END

```



```
END

--Recollim l'usuari i fer l'insert a logs
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('usuarisModificacio', @usu, GETDATE(), CAST(@usuari AS VARCHAR) + '~' + @nom +
'~' + @cognoms + '~' + CAST(@operador AS VARCHAR) + '~' + @email, @RSP)
END
GO

--
-- Definition for stored procedure altaDescarrega :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE dbo.altaDescarrega(
    @data varchar(12),
    @aplicacio varchar(10), @so varchar(10), @versio varchar(10),
    @usuari integer,
    @dispositiu integer,
    @mode integer,
    @wifi integer,
    @preu float OUTPUT,
    @RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @existeix integer
    DECLARE @operador integer
    DECLARE @pais varchar(2)
    DECLARE @IMEI varchar(17)

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    IF ((@data='') OR (@aplicacio='') OR (@so='') OR (@versio='') OR (@usuari='') OR
    (@dispositiu='') OR (@mode=''))
    BEGIN
        --Comprovem que s'han especificat tots els paràmetres
        SET @RSP='ERROR: Ha d'especificar tots els paràmetres d'entrada'
    END ELSE
    BEGIN
        --Cal comprovar que:
        --* aplicacio+so+versio sigui un codi vàlid de versió
        --* usuari sigui un codi vàlid d'usuari
        --* dispositiu sigui un codi vàlid de dispositiu
        --* mode sigui un codi vàlid de mode de pagament

        --Cal calcular el preu: buscar a versions_preus en el cas que sigui
        --el primer cop i sino 0
    
```

```
--Si wifi=0 aleshores 0 en qualsevol altre cas 1

        SELECT @existeix = COUNT(*)
FROM versions WHERE aplicacio = @aplicacio AND so = @so AND versio = @versio
    IF (@existeix=0)
        BEGIN
            --No existeix l'aplicació
            SET @RSP='ERROR: No existeix una aplicació amb aquest codi'
        END ELSE
        BEGIN
            SELECT @existeix = COUNT(*) FROM usuaris WHERE usuari = @usuari
            IF (@existeix = 0)
                BEGIN
                    --No existeix l'usuari
                    SET @RSP='ERROR: No existeix un usuari amb aquest codi'
                END ELSE
                BEGIN
                    SELECT @operador = COUNT(*) FROM usuaris WHERE usuari =
@usuari

                    SELECT @existeix = COUNT(*) FROM dispositius WHERE
dispositiu = @dispositiu
                    IF (@existeix = 0)
                        BEGIN
                            --No existeix el dispositiu
                            SET @RSP='ERROR: No existeix un dispositiu amb aquest codi'
                        END ELSE
                        BEGIN
                            SELECT @pais = pais, @IMEI = IMEI FROM dispositius
WHERE dispositiu = @dispositiu

                            SELECT @existeix = COUNT(*) FROM modes_pagament
WHERE mode = @mode
                            IF (@existeix=0)
                                BEGIN
                                    --No existeix el mode de pagament
                                    SET @RSP='ERROR: No existeix un mode de pagament
amb aquest codi'
                                END ELSE
                                BEGIN
                                    SELECT @existeix = COUNT(*)
FROM descarregues WHERE aplicacio = @aplicacio AND so
= @so AND versio = @versio
                                    AND usuari = @usuari
                                    IF (@existeix>1)
                                        BEGIN
                                            SET @preu=0
                                        END ELSE
                                        BEGIN
                                            SELECT @preu = preu
FROM versions_preus WHERE aplicacio = @aplicacio
AND so = @so AND versio = @versio
                                        END

                                        BEGIN TRY
                                            --el codi és autoincrement així que no cal incloure'l en l'insert
```

```

INSERT INTO descarregues (data, aplicacio, so,
versio, usuari, operador,
    pais, dispositiu, IMEI, preu, mode, wifi)
VALUES (@data, @aplicacio, @so, @versio, @usuari, @operador,
    @pais, @dispositiu, @IMEI, @preu, @mode, CASE WHEN @wifi=0 THEN 0
ELSE 1 END)

DECLARE @any varchar(4)
SELECT @any = YEAR(@data)
DECLARE @R varchar(255)
EXEC estadistica1 @RSP=@R OUTPUT
SET @RSP = @R + ','
EXEC estadistica2 @RSP=@R OUTPUT
SET @RSP = @R + ','
EXEC estadistica3 @any, @RSP=@R OUTPUT
SET @RSP = @R + ','
EXEC estadistica4 @any, @RSP=@R
OUTPUT
SET @RSP = @R + ','
EXEC estadistica5 @any, @RSP=@R
OUTPUT
SET @RSP = @R + ','
EXEC estadistica6 @any, @pais,
    @RSP=@R OUTPUT
SET @RSP = @R + ','
EXEC estadistica7 @any, @pais,
    @RSP=@R OUTPUT
SET @RSP = @R + ','
EXEC estadistica8 @any, @pais,
    @RSP=@R OUTPUT
SET @RSP = @R

END TRY
BEGIN CATCH
--Si l'insert no fos satisfactori retornarem el
missatge d'error
SET @RSP='ERROR: ' +
    ERROR_MESSAGE()
END CATCH;
END
END
END
END

--Recollim l'usuari i fer l'insert a logs
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('altaDescarrega', @usu, GETDATE(), CAST(@data AS VARCHAR) + '~' +
    CAST(@aplicacio AS VARCHAR) + '~' + CAST(@so AS VARCHAR) + '~' + CAST(@versio AS
    VARCHAR) + '~' + CAST(@usuari AS VARCHAR) + '~' + CAST(@dispositiu AS VARCHAR) + '~' +
    CAST(@mode AS VARCHAR) + '~' + CAST(@wifi AS VARCHAR), CAST(@preu AS
    VARCHAR), @RSP)
END
GO

```

```
--
-- Definition for stored procedure consulta1 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Per cada desenvolupador de un determinat país, tant les seves dades
--com el nombre d'aplicacions en les que ha participat
CREATE PROCEDURE dbo.consulta1(@pais varchar(2), @RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    IF (@pais='')
    BEGIN
        --Comprovem que s'ha especificat el codi de país
        SET @RSP='ERROR: No ha especificat el codi de país '
    END ELSE
    BEGIN
        --Mirem si existeix un país amb aquest codi
        SELECT @existeix = COUNT(*) FROM paisos WHERE pais = @pais
        IF (@existeix>0)
        BEGIN
            SELECT d.desenvolupador as Desenvolupador,
                d.nom as Nom,
                e.nom as Empresa,
                d.representant as Representant,
                d.adreca as 'Adreça',
                d.telefon as 'Telèfon',
                d.email as 'E-mail',
                p.nom as 'País',
                (SELECT COUNT(*)
                 FROM versions_desenvolupadors v
                 WHERE v.desenvolupador = d.desenvolupador
                 ) as 'Aplicacions'
            FROM desenvolupadors d
            LEFT JOIN desenvolupadors_empresa e ON (e.empresa = d.empresa)
            LEFT JOIN paisos p ON (p.pais = d.pais)
            WHERE p.pais = @pais
        END ELSE
        BEGIN
            --Si no existeix un país amb aquest codi retornem un error
            SET @RSP='ERROR: No existeix un país amb aquest codi'
        END
    END
END

--Recollim l'usuari i fer l'insert a logs
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
```

```
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('consulta1',@usu,GETDATE()),@pais,"",@RSP)
END
GO

--
-- Definition for stored procedure consulta2 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Dades principals d'aquelles aplicacions actives i el nombre de descàrregues
--fetes en ordre descendent, és a dir la primera aplicació de la llista serà
--la que més descàrregues hagi tingut
CREATE PROCEDURE dbo.consulta2(@RSP varchar(255) OUTPUT)
AS
BEGIN
    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'

    SELECT aplicacio as 'Aplicació', so as 'SO', versio as 'Versió',
        data_pujada as 'Data Pujada', binari as 'Enllaç Binari', mida as Mida,
        CAST(wminim as VARCHAR) + 'x' + CAST(hminim as VARCHAR) as 'Resolució Mínima',
        descarregues as 'Descàrregues'
    FROM (
        SELECT a.nom as aplicacio,
            s.nom as so,
            v.versio as versio,
            v.data_pujada,
            v.binari,
            v.mida,
            v.wminim,
            v.hminim,
            (SELECT COUNT(*)
                FROM descarregues d
                WHERE d.aplicacio = a.aplicacio
                AND d.so = s.so
                AND d.versio = v.versio) as descarregues
        FROM versions v
        LEFT JOIN aplicacions a ON (a.aplicacio = v.aplicacio)
        LEFT JOIN so s ON (s.so = v.so)
        WHERE v.actiu=1
    ) t
    ORDER BY descarregues DESC, aplicacio, so, versio

    --Recollim l'usuari i fer l'insert a logs
    DECLARE @usu varchar(30)
    SELECT @usu = SYSTEM_USER
    INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
    VALUES ('consulta2',@usu,GETDATE()),",",@RSP)
END
GO

--
```

```
-- Definition for stored procedure consulta3 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Llistat de països i nombre de descàrregues que ha tingut una determinada
--aplicació en un any concret
CREATE PROCEDURE dbo.consulta3(@aplicacio varchar(10), @so as varchar(10), @versio as
varchar(10),
    @any varchar(4), @RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    IF ((@aplicacio='') OR (@so='') OR (@versio='') OR (@any=''))
    BEGIN
        --Comprovem que els paràmetres venen plens
        SET @RSP='ERROR: Ha d'especificar l'aplicació i l'any dels que vol fer la
consulta '
    END ELSE
    BEGIN
        --Mirem si existeix l'aplicacio
        SELECT @existeix = COUNT(*) FROM versions
        WHERE aplicacio = @aplicacio AND so = @so AND versio = @versio
        IF (@existeix>0)
        BEGIN
            SELECT d.pais as 'País', p.nom as 'Nom', COUNT(*) as 'Descàrregues'
            FROM descarregues d
            LEFT JOIN paisos p ON (p.pais = d.pais)
            WHERE d.aplicacio = @aplicacio
            AND d.so = @so
            AND d.versio = @versio
            AND YEAR(d.data) = @any
            GROUP BY d.pais, p.nom
        END ELSE
        BEGIN
            --Si no existeix l'aplicacio retornem un error
            SET @RSP='ERROR: No existeix una aplicació amb aquest codi'
        END
    END
END

--Recollim l'usuari i fer l'insert a logs
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('consulta3',@usu,GETDATE(),@aplicacio + '~' + @so + '~' + @versio + '~' +
@any,',@RSP)
END
GO
```

```
--
-- Definition for stored procedure consulta4 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Llista de descàrregues d'un determinat usuari (identificat pel número de mòbil)
--on la primera és la més actual
CREATE PROCEDURE dbo.consulta4(@mobil varchar(20), @RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @usuari integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @usuari=0

    IF (@@mobil='')
    BEGIN
        --Comprovem que s'ha especificat el número de mòbil
        SET @RSP='ERROR: Ha d'especificar el número de mòbil del que vol fer la
consulta '
    END ELSE
    BEGIN
        --Mirem si existeix un usuari amb aquest número
        SELECT @usuari = usuari FROM usuaris WHERE mobil = @mobil
        IF (@usuari<>0)
        BEGIN
            SELECT d.data as Data,
                   d.aplicacio as 'Aplicació',
                   d.so as 'Sistema Operatiu',
                   d.versio as 'Versió',
                   d.preu as 'Preu',
                   m.nom as 'Mode Pag.',
                   CASE WHEN d.wifi = 0 THEN 'No' ELSE 'Sí' END as Wifi
            FROM descarregues d
            LEFT JOIN usuaris u ON (u.usuari = d.usuari)
            LEFT JOIN modes_pagament m ON (m.mode = d.mode)
            WHERE d.usuari = @usuari

            ORDER BY d.data DESC
        END ELSE
        BEGIN
            --Si no existeix l'usuari retornem un error
            SET @RSP='ERROR: No existeix un usuari amb aquest mòbil'
        END
    END

    --Recollim l'usuari i fer l'insert a logs
    DECLARE @usu varchar(30)
    SELECT @usu = SYSTEM_USER
    INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
    VALUES ('consulta4',@usu,GETDATE(),@mobil,@RSP)
END
GO
```

```
--
-- Definition for stored procedure consulta5 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Llista dels vint usuaris que més han gastat en aplicacions mòbil en un any concret,
--on el primer de la llista és l'usuari que més ha gastat
CREATE PROCEDURE dbo.consulta5(@any varchar(4), @RSP varchar(255) OUTPUT)
AS
BEGIN
    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'

    IF (@any='')
    BEGIN
        --Comprovem que s'ha especificat l'any
        SET @RSP='ERROR: Ha d'especificar l'any del que vol fer la consulta '
    END ELSE
    BEGIN
        SELECT TOP 20 * FROM (
            SELECT d.usuari, u.nom, u.mobil, SUM(preu) as preu
            FROM descarregues d
            LEFT JOIN usuaris u ON (u.usuari = d.usuari)
            WHERE YEAR(d.data) = @any
            GROUP BY d.usuari, u.nom, u.mobil
        ) t
        ORDER BY preu DESC
    END

    --Recollim l'usuari i fer l'insert a logs
    DECLARE @usu varchar(30)
    SELECT @usu = SYSTEM_USER
    INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
    VALUES ('consulta5',@usu,GETDATE(),@any,',',@RSP)
END
GO

--
-- Definition for stored procedure estadistica1 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--
CREATE PROCEDURE dbo.estadistica1(@RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @desc varchar(10)

    --Per defecte suposem que el procedure acabarà bé
```



```
SET @RSP='OK'

SELECT @desc = COUNT(*) FROM descarregues
BEGIN TRY
    BEGIN TRAN
        DELETE FROM estadistiques WHERE estadistica = 1
    INSERT INTO estadistiques (estadistica, valor, data) VALUES (1, @desc, GETDATE())
    COMMIT TRAN
END TRY
    BEGIN CATCH
        --Si hi ha algun error al fer el delete o l'insert no es farà
        --cap dels dos i retornarem el missatge d'error
        ROLLBACK TRAN
        SET @RSP='ERROR: ' + ERROR_MESSAGE()
    END CATCH;

--Recollim l'usuari i fer l'insert a logs
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('estadistica1', @usu, GETDATE(), '', @RSP)
END
GO

--
-- Definition for stored procedure estadistica2 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Total d'euros ingressats en descarregues fetes fins al moment
CREATE PROCEDURE dbo.estadistica2(@RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @preu varchar(10)

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'

    SELECT @preu = SUM(preu) FROM descarregues
    BEGIN TRY
        BEGIN TRAN
            DELETE FROM estadistiques WHERE estadistica = 2
        INSERT INTO estadistiques (estadistica, valor, data) VALUES (2, @preu, GETDATE())
        COMMIT TRAN
    END TRY
        BEGIN CATCH
            --Si hi ha algun error al fer el delete o l'insert no es farà
            --cap dels dos i retornarem el missatge d'error
            ROLLBACK TRAN
            SET @RSP='ERROR: ' + ERROR_MESSAGE()
        END CATCH;

    --Recollim l'usuari i fer l'insert a logs
```

```
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('estadistica2', @usu, GETDATE(), '', @RSP)
END
GO

--
-- Definition for stored procedure estadistica3 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--Mitja d'aplicaciones descarregades per usuari en un any concret
CREATE PROCEDURE dbo.estadistica3(@any varchar(4), @RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @num_usu varchar(10)
    DECLARE @num_apl varchar(10)
    DECLARE @mitja varchar(10)

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'

    IF (@any='')
    BEGIN
        --Comprovem que el paràmetre de l'any ve ple
        SET @RSP='ERROR: Ha d'especificar l'any del que vol fer la consulta '
    END ELSE
    BEGIN
        --Per a qualsevol usuari, actiu o no actiu
        SELECT @num_usu = COUNT(d.usuari)
        FROM descarregues d
        WHERE YEAR(d.data) = @any
        GROUP BY d.usuari

        SELECT @num_apl = COUNT(d.aplicacio)
        FROM descarregues d
        WHERE YEAR(d.data) = @any
        GROUP BY d.aplicacio

        SELECT @mitja = CAST (@num_usu AS FLOAT) / CAST(@num_apl AS FLOAT)

        BEGIN TRY
        BEGIN TRAN
            DELETE FROM estadistiques WHERE param = @any AND estadistica = 3
            INSERT INTO estadistiques (estadistica, valor, param, data) VALUES (3, @mitja,
@any, GETDATE())
        COMMIT TRAN
        END TRY
        BEGIN CATCH
            --Si hi ha algun error al fer el delete o l'insert no es farà
            --cap dels dos i retornarem el missatge d'error
            ROLLBACK TRAN
        END CATCH
    END
END
```

```
                SET @RSP='ERROR: ' + ERROR_MESSAGE()
            END CATCH;
        END

        --Recollim l'usuari i fer l'insert a logs
        DECLARE @usu varchar(30)
        SELECT @usu = SYSTEM_USER
        INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
        VALUES ('estadistica3', @usu, GETDATE(), @any, @RSP)
    END
GO

--
-- Definition for stored procedure estadistica4 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--
CREATE PROCEDURE dbo.estadistica4(@any varchar(4), @RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @des varchar(10)

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'

    IF (@any='')
    BEGIN
        --Comprovem que el paràmetre de l'any ve ple
        SET @RSP='ERROR: Ha d'especificar l'any del que vol fer la consulta '
    END ELSE
    BEGIN
        SELECT TOP 1 @des = CAST(desenvolupador as VARCHAR) + '~' + CAST(num
as VARCHAR)
        FROM (
            SELECT t.desenvolupador, SUM(num) as num
            FROM (
                SELECT
                    d.aplicacio, d.so, d.versio, vd.desenvolupador, COUNT(*) as num
                FROM descarregues d
                LEFT JOIN versions v ON (v.aplicacio = d.aplicacio AND v.so =
d.so AND v.versio = d.versio)
                LEFT JOIN versions_desenvolupadors vd ON (vd.aplicacio =
d.aplicacio AND vd.so = d.so AND vd.versio = d.versio)
                WHERE YEAR(d.data) = @any
                GROUP BY d.aplicacio, d.so, d.versio, vd.desenvolupador
            ) t
            GROUP BY t.desenvolupador
        ) u
        ORDER BY num DESC

    BEGIN TRY
        BEGIN TRAN
```

```
        DELETE FROM estadistiques WHERE param = @any AND estadistica = 4
        INSERT INTO estadistiques (estadistica, valor, param, data) VALUES (4, @des,
@any, GETDATE())
        COMMIT TRAN
        END TRY
        BEGIN CATCH
            --Si hi ha algun error al fer el delete o l'insert no es farà
            --cap dels dos i retornarem el missatge d'error
            ROLLBACK TRAN
            SET @RSP='ERROR: ' + ERROR_MESSAGE()
        END CATCH;
    END

    --Recollim l'usuari i fer l'insert a logs
    DECLARE @usu varchar(30)
    SELECT @usu = SYSTEM_USER
    INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
    VALUES ('estadistica4', @usu, GETDATE(), @any, '', @RSP)
END
GO

--
-- Definition for stored procedure estadistica5 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--
CREATE PROCEDURE dbo.estadistica5(@any varchar(4), @RSP varchar(255) OUTPUT)
AS
BEGIN
    DECLARE @res varchar(100)
    DECLARE @apl varchar(10)
    DECLARE @so varchar(10)
    DECLARE @versio varchar(10)
    DECLARE @preu varchar(10)

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'

    IF (@any='')
    BEGIN
        --Comprovem que el paràmetre de l'any ve ple
        SET @RSP='ERROR: Ha d"especificar l"any del que vol fer la consulta '
    END ELSE
    BEGIN
        SELECT TOP 1 @apl = aplicacio, @so = so, @versio = versio, @preu = preu
        FROM (
            SELECT d.aplicacio, d.so, d.versio, SUM(d.preu) as preu
            FROM descarregues d
            WHERE YEAR(d.data) = @any
            GROUP BY d.aplicacio, d.so, d.versio
        ) t
        ORDER BY preu DESC
    END
END
```

```
        SELECT @res = STUFF(
        (SELECT CAST('#' AS VARCHAR(MAX)) + fa
        FROM (
            SELECT CAST(d.desenvolupador AS VARCHAR) AS fa
            FROM versions_desenvolupadors d
            WHERE d.aplicacio = @apl AND d.so = @so AND d.versio = @versio
            GROUP BY d.desenvolupador
        ) t
        GROUP BY fa
        FOR XML PATH(''),1,1,")

        SET @res = @apl + '~' + @so + '~' + @versio + '~' + @preu + '~' + @res

    BEGIN TRY
        BEGIN TRAN
            DELETE FROM estadistiques WHERE param = @any AND estadistica = 5
            INSERT INTO estadistiques (estadistica, valor, param, data) VALUES (5, @res,
@any, GETDATE())
        COMMIT TRAN
    END TRY
    BEGIN CATCH
        --Si hi ha algun error al fer el delete o l'insert no es farà
        --cap dels dos i retornarem el missatge d'error
        ROLLBACK TRAN
        SET @RSP='ERROR: ' + ERROR_MESSAGE()
    END CATCH;

END

--Recollim l'usuari i fer l'insert a logs
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('estadistica5', @usu, GETDATE(), @any, '', @RSP)
END
GO

--
-- Definition for stored procedure estadistica6 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--
CREATE PROCEDURE dbo.estadistica6(@any varchar(4), @pais varchar(2), @RSP varchar(255)
OUTPUT)
AS
BEGIN
    DECLARE @res varchar(100)
    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0
```

```
IF ((@any='') OR (@pais=''))
BEGIN
    --Comprovem que els paràmetres venen plens
    SET @RSP='ERROR: Ha d'especificar l'any i el país del que vol fer la consulta '
END ELSE
BEGIN
    --Mirem si existeix un país amb aquest codi
    SELECT @existeix = COUNT(*) FROM paisos WHERE pais = @pais
    IF (@existeix>0)
    BEGIN
        SELECT @res = COUNT(DISTINCT d.usuari)
        FROM descarregues d
        WHERE YEAR(d.data) = @any
        AND d.pais = @pais
        --Per a qualsevol usuari, actiu o no actiu

        BEGIN TRY
            BEGIN TRAN
                DELETE FROM estadistiques WHERE param = @any + '~' +
                @pais AND estadistica = 6
                INSERT INTO estadistiques (estadistica, valor, param, data) VALUES (6,
                @res, @any + '~' + @pais, GETDATE())
            COMMIT TRAN
        END TRY
        BEGIN CATCH
            --Si hi ha algun error al fer el delete o l'insert no es farà
            --cap dels dos i retornarem el missatge d'error
            ROLLBACK TRAN
            SET @RSP='ERROR: ' + ERROR_MESSAGE()
        END CATCH;
    END ELSE
    BEGIN
        --Si no existeix un país amb aquest codi retornem un error
        SET @RSP='ERROR: No existeix un país amb aquest codi'
    END
END

--Recollim l'usuari i fer l'insert a logs
DECLARE @usu varchar(30)
SELECT @usu = SYSTEM_USER
INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
VALUES ('estadistica6', @usu, GETDATE(), @any + '~' + @pais, '', @RSP)
END
GO

--
-- Definition for stored procedure estadistica7 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO
--
```

```
CREATE PROCEDURE dbo.estadistica7(@any varchar(4), @pais varchar(2), @RSP varchar(255)
OUTPUT)
AS
BEGIN
    DECLARE @res varchar(100)
    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    IF ((@any='') OR (@pais=''))
    BEGIN
        --Comprovem que els paràmetres venen plens
        SET @RSP='ERROR: Ha d'especificar l'any i el país del que vol fer la consulta '
    END ELSE
    BEGIN
        --Mirem si existeix un país amb aquest codi
        SELECT @existeix = COUNT(*) FROM paisos WHERE pais = @pais
        IF (@existeix>0)
        BEGIN
            SELECT @res = SUM(d.preu)
            FROM descarregues d
            WHERE YEAR(d.data) = @any
            AND d.pais = @pais
            --Per a qualsevol usuari, actiu o no actiu

            BEGIN TRY
                BEGIN TRAN
                DELETE FROM estadistiques WHERE param = @any + '~' +
@pais AND estadistica = 7
                INSERT INTO estadistiques (estadistica, valor, param, data) VALUES (7,
@res, @any + '~' + @pais, GETDATE())
                COMMIT TRAN
            END TRY
            BEGIN CATCH
                --Si hi ha algun error al fer el delete o l'insert no es farà
                --cap dels dos i retornarem el missatge d'error
                ROLLBACK TRAN
                SET @RSP='ERROR: ' + ERROR_MESSAGE()
            END CATCH;
        END ELSE
        BEGIN
            --Si no existeix un país amb aquest codi retornem un error
            SET @RSP='ERROR: No existeix un país amb aquest codi'
        END
    END

    --Recollim l'usuari i fer l'insert a logs
    DECLARE @usu varchar(30)
    SELECT @usu = SYSTEM_USER
    INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
    VALUES ('estadistica7',@usu,GETDATE(),@any + '~' + @pais,@RSP)
END
GO
```

```
--
-- Definition for stored procedure estadistica8 :
--
GO
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

--
CREATE PROCEDURE dbo.estadistica8(@any varchar(4), @pais varchar(2), @RSP varchar(255)
OUTPUT)
AS
BEGIN
    DECLARE @res varchar(100)
    DECLARE @existeix integer

    --Per defecte suposem que el procedure acabarà bé
    SET @RSP='OK'
    SET @existeix=0

    IF ((@any='') OR (@pais=''))
    BEGIN
        --Comprovem que els paràmetres venen plens
        SET @RSP='ERROR: Ha d'especificar l'any i el pais del que vol fer la consulta '
    END ELSE
    BEGIN
        --Mirem si existeix un país amb aquest codi
        SELECT @existeix = COUNT(*) FROM paisos WHERE pais = @pais
        IF (@existeix>0)
        BEGIN
            SELECT @res = COUNT(*)

            FROM (
                SELECT aplicacio
                FROM descarregues
                WHERE YEAR(data) = @any
                AND pais = @pais
                GROUP BY aplicacio
            ) t

            BEGIN TRY
                BEGIN TRAN
                    DELETE FROM estadistiques WHERE param = @any + '~' +
@pais AND estadistica = 8
                    INSERT INTO estadistiques (estadistica, valor, param, data) VALUES (8,
@res, @any + '~' + @pais, GETDATE())
                COMMIT TRAN
            END TRY
            BEGIN CATCH
                --Si hi ha algun error al fer el delete o l'insert no es farà
                --cap dels dos i retornarem el missatge d'error
                ROLLBACK TRAN
                SET @RSP='ERROR: ' + ERROR_MESSAGE()
            END CATCH;
        END ELSE
        BEGIN
            --Si no existeix un país amb aquest codi retornem un error
        END
    END
END
```



```
                SET @RSP='ERROR: No existeix un país amb aquest codi'
            END
        END

        --Recollim l'usuari i fer l'insert a logs
        DECLARE @usu varchar(30)
        SELECT @usu = SYSTEM_USER
        INSERT INTO logs (procediment, usuari_bd, instant, entrada, sortida, rsp)
        VALUES ('estadistica8', @usu, GETDATE(), @any + '-' + @pais, @RSP)
    END
GO
```

ANNEX 4. ALTRES INSERTS

```
--
-- Data for table dbo.aplicacions (LIMIT 0,500)
--

INSERT INTO [dbo].[aplicacions] ([aplicacio], [nom])
VALUES
    (N'APL1', N'APLICACIÓ 1')
GO

INSERT INTO [dbo].[aplicacions] ([aplicacio], [nom])
VALUES
    (N'APL2', N'APLICACIÓ 2')
GO

INSERT INTO [dbo].[aplicacions] ([aplicacio], [nom])
VALUES
    (N'APL3', N'APLICACIÓ 3')
GO

INSERT INTO [dbo].[aplicacions] ([aplicacio], [nom])
VALUES
    (N'APL4', N'APLICACIÓ 4')
GO

INSERT INTO [dbo].[aplicacions] ([aplicacio], [nom])
VALUES
    (N'APL5', N'APLICACIÓ 5')
GO

INSERT INTO [dbo].[aplicacions] ([aplicacio], [nom])
VALUES
    (N'APL6', N'APLICACIÓ 6')
GO

INSERT INTO [dbo].[aplicacions] ([aplicacio], [nom])
VALUES
    (N'APL7', N'APLICACIÓ 7')
GO

INSERT INTO [dbo].[aplicacions] ([aplicacio], [nom])
VALUES
```

```
(N'APL8', N'APLICACIÓ 8')
GO
```

```
INSERT INTO [dbo].[aplicacions] ([aplicacio], [nom])
VALUES
(N'APL9', N'APLICACIÓ 9')
GO
```

```
--
-- Data for table dbo.descripcion (LIMIT 0,500)
--
```

```
INSERT INTO [dbo].[descripcion] ([aplicacio], [idioma], [descripcion])
VALUES
(N'APL1', N'CA', N'DESCRIPCIÓ APLICACIÓ 1')
GO
```

```
INSERT INTO [dbo].[descripcion] ([aplicacio], [idioma], [descripcion])
VALUES
(N'APL1', N'ES', N'DESCRIPCIÓN APLICACIÓN 1')
GO
```

```
INSERT INTO [dbo].[descripcion] ([aplicacio], [idioma], [descripcion])
VALUES
(N'APL2', N'CA', N'DESCRIPCIÓ APLICACIÓ 2')
GO
```

```
INSERT INTO [dbo].[descripcion] ([aplicacio], [idioma], [descripcion])
VALUES
(N'APL2', N'ES', N'DESCRIPCIÓN APLICACIÓN 2')
GO
```

```
INSERT INTO [dbo].[descripcion] ([aplicacio], [idioma], [descripcion])
VALUES
(N'APL3', N'CA', N'DESCRIPCIÓ APLICACIÓ 3')
GO
```

```
INSERT INTO [dbo].[descripcion] ([aplicacio], [idioma], [descripcion])
VALUES
(N'APL4', N'CA', N'DESCRIPCIÓ APLICACIÓ 4')
GO
```

```
INSERT INTO [dbo].[descripcion] ([aplicacio], [idioma], [descripcion])
VALUES
(N'APL5', N'CA', N'DESCRIPCIÓ APLICACIÓ 5')
GO
```

```
INSERT INTO [dbo].[descripcion] ([aplicacio], [idioma], [descripcion])
VALUES
(N'APL6', N'CA', N'DESCRIPCIÓ APLICACIÓ 6')
GO
```

```
INSERT INTO [dbo].[descripcion] ([aplicacio], [idioma], [descripcion])
VALUES
(N'APL7', N'CA', N'DESCRIPCIÓ APLICACIÓ 7')
GO
```

```
INSERT INTO [dbo].[descripcions] ([aplicacio], [idioma], [descripcio])
VALUES
(N'APL8', N'CA', N'DESCRIPCIÓ APLICACIÓ 8')
GO

INSERT INTO [dbo].[descripcions] ([aplicacio], [idioma], [descripcio])
VALUES
(N'APL9', N'CA', N'DESCRIPCIÓ APLICACIÓ 9')
GO

--
-- Data for table dbo.desenvolupadors_empreses (LIMIT 0,500)
--

INSERT INTO [dbo].[desenvolupadors_empreses] ([empresa], [nom])
VALUES
(1, N'EMPRESA 1')
GO

INSERT INTO [dbo].[desenvolupadors_empreses] ([empresa], [nom])
VALUES
(2, N'EMPRESA 2')
GO

INSERT INTO [dbo].[desenvolupadors_empreses] ([empresa], [nom])
VALUES
(3, N'EMPRESA 3')
GO

INSERT INTO [dbo].[desenvolupadors_empreses] ([empresa], [nom])
VALUES
(4, N'EMPRESA 4')
GO

INSERT INTO [dbo].[desenvolupadors_empreses] ([empresa], [nom])
VALUES
(5, N'EMPRESA 5')
GO

INSERT INTO [dbo].[desenvolupadors_empreses] ([empresa], [nom])
VALUES
(6, N'EMPRESA 6')
GO

INSERT INTO [dbo].[desenvolupadors_empreses] ([empresa], [nom])
VALUES
(7, N'EMPRESA 7')
GO

INSERT INTO [dbo].[desenvolupadors_empreses] ([empresa], [nom])
VALUES
(8, N'EMPRESA 8')
GO

INSERT INTO [dbo].[desenvolupadors_empreses] ([empresa], [nom])
```

```
VALUES
(9, N'EMPRESA 9')
GO

--
-- Data for table dbo.dispositius (LIMIT 0,500)
--

INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(1, N'ESXXXX01ZZZZZ10', 240, 320, N'001001', N'ES')
GO

INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(2, N'ESXXXX01ZZZZZ20', 240, 320, N'001001', N'ES')
GO

INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(3, N'ESXXXX01ZZZZZ30', 240, 320, N'001002', N'ES')
GO

INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(4, N'ESXXXX01ZZZZZ40', 240, 320, N'001003', N'ES')
GO

INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(5, N'ESXXXX05ZZZZZ50', 240, 320, N'005001', N'ES')
GO

INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(6, N'ESXXXX05ZZZZZ60', 240, 320, N'005001', N'ES')
GO

INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(7, N'ESXXXX05ZZZZZ70', 240, 320, N'005002', N'ES')
GO

INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(8, N'ESXXXX05ZZZZZ80', 240, 320, N'005002', N'ES')
GO

INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(9, N'ESXXXX06ZZZZZ90', 240, 320, N'006001', N'ES')
GO

INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(10, N'ESXXXX07ZZZZ100', 240, 320, N'007001', N'ES')
```

GO

```
INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(11, N'ESXXX07ZZZZ110', 240, 320, N'007002', N'ES')
GO
```

```
INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(12, N'ESXXX08ZZZZ120', 240, 320, N'008001', N'ES')
GO
```

```
INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(13, N'ESXXX08ZZZZ130', 240, 320, N'008002', N'ES')
GO
```

```
INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(14, N'ESXXX09ZZZZ140', 240, 320, N'009001', N'ES')
GO
```

```
INSERT INTO [dbo].[dispositius] ([dispositiu], [IMEI], [w], [h], [model], [pais])
VALUES
(15, N'ESXXX09ZZZZ150', 240, 320, N'009002', N'ES')
GO
```

ANNEX 5. PROVES

```
DECLARE @R1 varchar(255)
EXEC aplicacioAlta 'APL1','AND','1','enllaç binari',10,200,300,1,'url',1.5, @RSP=@R1 OUTPUT
PRINT @R1
```

```
DECLARE @R2 varchar(255)
EXEC aplicacioAlta 'APL1','AND','1','enllaç binari',10,200,300,1,'url',1.5, @RSP=@R2 OUTPUT
PRINT @R2
```

```
DECLARE @R3 varchar(255)
EXEC aplicacioAlta 'APL2','AND','1','enllaç binari',10,200,300,1,'url',1, @RSP=@R3 OUTPUT
PRINT @R3
```

```
INSERT INTO [dbo].[versions_url] ([aplicacio], [so], [versio], [idioma], [url])
VALUES ('APL2', 'AND', '1', 'ES', 'url2')
GO
```

```
INSERT INTO [dbo].[versions_preus] ([aplicacio], [so], [versio], [pais], [preu])
VALUES ('APL2', 'AND', '1', 'GB', 1.2)
GO
```

```
DECLARE @R4 varchar(255)
EXEC aplicacioAlta 'APL2','IOS','1','enllaç binari',10,200,300,1,'url',1, @RSP=@R4 OUTPUT
PRINT @R4
```

```
DECLARE @R5 varchar(255)
EXEC aplicacioAlta 'APL3','IOS','1','enllaç binari',10,200,300,1,'url',1, @RSP=@R5 OUTPUT
```

PRINT @R5

DECLARE @R6 varchar(255)
EXEC aplicacioAlta 'APL3','IOS','2','enllaç binari',10,200,300,1,'url',1, @RSP=@R6 OUTPUT
PRINT @R6

DECLARE @R7 varchar(255)
EXEC aplicacioBaixa 'APL1','AND','1', @RSP=@R7 OUTPUT
PRINT @R7

DECLARE @R8 varchar(255)
EXEC aplicacioBaixa 'APL2','AND','1', @RSP=@R8 OUTPUT
PRINT @R8

DECLARE @R9 varchar(255)
EXEC aplicacioBaixa 'APL2','AND','1', @RSP=@R9 OUTPUT
PRINT @R9

DECLARE @R10 varchar(255)
EXEC aplicacioModificacio 'APL2','AND','1', @RSP=@R10 OUTPUT
PRINT @R10

DECLARE @R11 varchar(255)
EXEC aplicacioModificacio 'APL2','IOS','1', @RSP=@R11 OUTPUT
PRINT @R11

DECLARE @R12 varchar(255)
EXEC aplicacioModificacio 'APL2','BB','1', @RSP=@R12 OUTPUT
PRINT @R12

DECLARE @R13 varchar(255)
EXEC usuariAlta 1,'NOM 1','COGNOMS 1','600000001', 1, '1@uoc.edu', @RSP=@R13 OUTPUT
PRINT @R13

DECLARE @R14 varchar(255)
EXEC usuariAlta 2,'NOM 2','COGNOMS 2','600000001', 1, '2@uoc.edu', @RSP=@R14 OUTPUT
PRINT @R14

DECLARE @R15 varchar(255)
EXEC usuariAlta 3,'NOM 3','COGNOMS 3','600000003', 1, '3@uoc.edu', @RSP=@R15 OUTPUT
PRINT @R15

DECLARE @R16 varchar(255)
EXEC usuariAlta 4,'NOM 4','COGNOMS 4','600000004', 6, '4@uoc.edu', @RSP=@R16 OUTPUT
PRINT @R16

DECLARE @R17 varchar(255)
EXEC usuariAlta 5,'NOM 5','COGNOMS 5','600000005', 6, '5@uoc.edu', @RSP=@R17 OUTPUT
PRINT @R17

DECLARE @R18 varchar(255)
EXEC usuariAlta 6,'NOM 6','COGNOMS 6','600000006', 6, '6@uoc.edu', @RSP=@R18 OUTPUT
PRINT @R18

DECLARE @R19 varchar(255)
EXEC usuariAlta 7,'NOM 7','COGNOMS 7','600000007', 6, '7@uoc.edu', @RSP=@R19 OUTPUT

PRINT @R19

DECLARE @R20 varchar(255)
EXEC usuariosAlta 8,'NOM 8','COGNOMS 8','600000008', 6, '8@uoc.edu', @RSP=@R20 OUTPUT
PRINT @R20

DECLARE @R21 varchar(255)
EXEC usuariosAlta 9,'NOM 9','COGNOMS 9','600000009', 10, '9@uoc.edu', @RSP=@R21
OUTPUT
PRINT @R21

DECLARE @R22 varchar(255)
EXEC usuariosAlta 10,'NOM 10','COGNOMS 10','600000010', 10, '00@uoc.edu', @RSP=@R22
OUTPUT
PRINT @R22

DECLARE @R23 varchar(255)
EXEC usuariosBaixa ", @RSP=@R23 OUTPUT
PRINT @R23

DECLARE @R24 varchar(255)
EXEC usuariosBaixa 30, @RSP=@R24 OUTPUT
PRINT @R24

DECLARE @R25 varchar(255)
EXEC usuariosBaixa 8, @RSP=@R25 OUTPUT
PRINT @R25

DECLARE @R26 varchar(255)
EXEC usuariosModificacio 10, ", ", 0, '10@uoc.edu', @RSP=@R26 OUTPUT
PRINT @R26

DECLARE @R27 varchar(255)
EXEC desarrolladorAlta 1,'NOM I COGNOMS 1', 1, 'REPRESENTANT 1', 'ADREÇA 1',
'660000001', 'd1@uoc.edu', 'ES', @RSP=@R27 OUTPUT
PRINT @R27

DECLARE @R28 varchar(255)
EXEC desarrolladorAlta 2,'NOM I COGNOMS 2', 1, 'REPRESENTANT 1', 'ADREÇA 2',
'660000002', 'd2@uoc.edu', 'ES', @RSP=@R28 OUTPUT
PRINT @R28

DECLARE @R29 varchar(255)
EXEC desarrolladorAlta 3,'NOM I COGNOMS 3', 0, ", 'ADREÇA 3', '660000003', 'd3@uoc.edu',
'ES', @RSP=@R29 OUTPUT
PRINT @R29

DECLARE @R30 varchar(255)
EXEC desarrolladorAlta 4,'NOM I COGNOMS 4', 0, ", 'ADREÇA 4', '660000004', 'd4@uoc.edu',
'ES', @RSP=@R30 OUTPUT
PRINT @R30

DECLARE @R31 varchar(255)
EXEC desarrolladorAlta 5,'NOM I COGNOMS 5', 0, ", 'ADREÇA 5', '660000005', 'd5@uoc.edu',
'ES', @RSP=@R31 OUTPUT
PRINT @R31

```
DECLARE @R32 varchar(255)
EXEC desenvolupadorAlta 6,'NOM I COGNOMS 6', 0, '', 'ADREÇA 6', '660000006', 'd6@uoc.edu',
'GB', @RSP=@R32 OUTPUT
PRINT @R32
```

```
DECLARE @R33 varchar(255)
EXEC desenvolupadorAlta 7,'NOM I COGNOMS 7', 0, '', 'ADREÇA 7', '660000007', 'd7@uoc.edu',
'GB', @RSP=@R33 OUTPUT
PRINT @R33
```

```
DECLARE @R34 varchar(255)
EXEC desenvolupadorAlta 8,'NOM I COGNOMS 8', 0, '', 'ADREÇA 8', '660000008', 'd8@uoc.edu',
'GB', @RSP=@R34 OUTPUT
PRINT @R34
```

```
DECLARE @R35 varchar(255)
EXEC desenvolupadorAlta 9,'NOM I COGNOMS 9', 0, '', 'ADREÇA 9', '660000009', 'd9@uoc.edu',
'FR', @RSP=@R35 OUTPUT
PRINT @R35
```

```
DECLARE @R36 varchar(255)
EXEC desenvolupadorAlta 10,'NOM I COGNOMS 10', 0, '', 'ADREÇA 99999999', '660000010',
'd10@uoc.edu', 'FR', @RSP=@R36 OUTPUT
PRINT @R36
```

```
DECLARE @R27 varchar(255)
EXEC desenvolupadorAlta 1,'NOM I COGNOMS 1', 1, 'REPRESENTANT 1', 'ADREÇA 1',
'660000001', 'd1@uoc.edu', 'ES', @RSP=@R27 OUTPUT
PRINT @R27
```

```
DECLARE @R28 varchar(255)
EXEC desenvolupadorAlta 2,'NOM I COGNOMS 2', 1, 'REPRESENTANT 1', 'ADREÇA 2',
'660000002', 'd2@uoc.edu', 'ES', @RSP=@R28 OUTPUT
PRINT @R28
```

```
DECLARE @R29 varchar(255)
EXEC desenvolupadorAlta 3,'NOM I COGNOMS 3', 0, '', 'ADREÇA 3', '660000003', 'd3@uoc.edu',
'ES', @RSP=@R29 OUTPUT
PRINT @R29
```

```
DECLARE @R30 varchar(255)
EXEC desenvolupadorAlta 4,'NOM I COGNOMS 4', 0, '', 'ADREÇA 4', '660000004', 'd4@uoc.edu',
'ES', @RSP=@R30 OUTPUT
PRINT @R30
```

```
DECLARE @R31 varchar(255)
EXEC desenvolupadorAlta 5,'NOM I COGNOMS 5', 0, '', 'ADREÇA 5', '660000005', 'd5@uoc.edu',
'ES', @RSP=@R31 OUTPUT
PRINT @R31
```

```
DECLARE @R32 varchar(255)
EXEC desenvolupadorAlta 6,'NOM I COGNOMS 6', 0, '', 'ADREÇA 6', '660000006', 'd6@uoc.edu',
'GB', @RSP=@R32 OUTPUT
PRINT @R32
```



```
DECLARE @R33 varchar(255)
EXEC desenvolupadorAlta 7,'NOM I COGNOMS 7', 0, '', 'ADREÇA 7', '660000007', 'd7@uoc.edu',
'GB', @RSP=@R33 OUTPUT
PRINT @R33
```

```
DECLARE @R34 varchar(255)
EXEC desenvolupadorAlta 8,'NOM I COGNOMS 8', 0, '', 'ADREÇA 8', '660000008', 'd8@uoc.edu',
'GB', @RSP=@R34 OUTPUT
PRINT @R34
```

```
DECLARE @R35 varchar(255)
EXEC desenvolupadorAlta 9,'NOM I COGNOMS 9', 0, '', 'ADREÇA 9', '660000009', 'd9@uoc.edu',
'FR', @RSP=@R35 OUTPUT
PRINT @R35
```

```
DECLARE @R36 varchar(255)
EXEC desenvolupadorAlta 10,'NOM I COGNOMS 10', 0, '', 'ADREÇA 99999999', '660000010',
'd10@uoc.edu', 'FR', @RSP=@R36 OUTPUT
PRINT @R36
```

```
DECLARE @R37 varchar(255)
EXEC desenvolupadorModificacio 30, "", 0, "", 'ADREÇA 10', "", "", @RSP=@R37 OUTPUT
PRINT @R37
```

```
DECLARE @R38 varchar(255)
EXEC desenvolupadorModificacio 10, "", 0, "", 'ADREÇA 10', "", "", @RSP=@R38 OUTPUT
PRINT @R38
```

```
INSERT INTO [dbo].[versions_desenvolupadors] ([aplicacio], [so], [versio], [desenvolupador])
VALUES ('APL1', 'AND', '1', 1)
GO
```

```
INSERT INTO [dbo].[versions_desenvolupadors] ([aplicacio], [so], [versio], [desenvolupador])
VALUES ('APL1', 'AND', '1', 10)
GO
```

```
INSERT INTO [dbo].[versions_desenvolupadors] ([aplicacio], [so], [versio], [desenvolupador])
VALUES ('APL2', 'AND', '1', 2)
GO
```

```
INSERT INTO [dbo].[versions_desenvolupadors] ([aplicacio], [so], [versio], [desenvolupador])
VALUES ('APL2', 'AND', '1', 3)
GO
```

```
INSERT INTO [dbo].[versions_desenvolupadors] ([aplicacio], [so], [versio], [desenvolupador])
VALUES ('APL2', 'AND', '1', 4)
GO
```

```
INSERT INTO [dbo].[versions_desenvolupadors] ([aplicacio], [so], [versio], [desenvolupador])
VALUES ('APL2', 'IOS', '1', 2)
GO
```

```
INSERT INTO [dbo].[versions_desenvolupadors] ([aplicacio], [so], [versio], [desenvolupador])
VALUES ('APL3', 'IOS', '1', 5)
GO
```

```
INSERT INTO [dbo].[versions_desenvolupadors] ([aplicacio], [so], [versio], [desenvolupador])  
VALUES ('APL3', 'IOS', '1', 6)  
GO
```

```
INSERT INTO [dbo].[versions_desenvolupadors] ([aplicacio], [so], [versio], [desenvolupador])  
VALUES ('APL3', 'IOS', '1', 7)  
GO
```

```
INSERT INTO [dbo].[versions_desenvolupadors] ([aplicacio], [so], [versio], [desenvolupador])  
VALUES ('APL3', 'IOS', '1', 9)  
GO
```

```
INSERT INTO [dbo].[versions_desenvolupadors] ([aplicacio], [so], [versio], [desenvolupador])  
VALUES ('APL3', 'IOS', '2', 5)  
GO
```

```
INSERT INTO [dbo].[versions_desenvolupadors] ([aplicacio], [so], [versio], [desenvolupador])  
VALUES ('APL3', 'IOS', '2', 6)  
GO
```

```
DECLARE @R39 varchar(255)  
EXEC desenvolupadorBaixa 8, @RSP=@R39 OUTPUT  
PRINT @R39
```

```
DECLARE @R40 varchar(255)  
EXEC desenvolupadorBaixa 10, @RSP=@R40 OUTPUT  
PRINT @R40
```

```
DECLARE @R41 varchar(255)  
DECLARE @p41 float  
EXEC altaDescarrega '15/1/2013','APL1', 'AND', '1', 1, 1, 1, 0, @preu=@p41 OUTPUT,  
@RSP=@R41 OUTPUT  
PRINT @R41  
PRINT @p41
```

```
DECLARE @R42 varchar(255)  
DECLARE @p42 float  
EXEC altaDescarrega '20/1/2013','APL1', 'AND', '1', 1, 1, 1, 0, @preu=@p42 OUTPUT,  
@RSP=@R42 OUTPUT  
PRINT @R42  
PRINT @p42
```

```
DECLARE @R43 varchar(255)  
DECLARE @p43 float  
EXEC altaDescarrega '15/1/2013','APL1', 'AND', '1', 1, 2, 1, 0, @preu=@p43 OUTPUT,  
@RSP=@R43 OUTPUT  
PRINT @R43  
PRINT @p43
```

```
DECLARE @R44 varchar(255)  
DECLARE @p44 float  
EXEC altaDescarrega '16/1/2013','APL1', 'AND', '1', 3, 3, 1, 0, @preu=@p44 OUTPUT,  
@RSP=@R44 OUTPUT  
PRINT @R44  
PRINT @p44
```

```
DECLARE @R45 varchar(255)
DECLARE @p45 float
EXEC altaDescarrega '16/1/2014','APL1', 'AND', '1', 4, 4, 1, 0, @preu=@p45 OUTPUT,
@RSP=@R45 OUTPUT
PRINT @R45
PRINT @p45
```

```
DECLARE @R46 varchar(255)
DECLARE @p46 float
EXEC altaDescarrega '1/2/2013','APL2', 'AND', '1', 4, 4, 1, 0, @preu=@p46 OUTPUT,
@RSP=@R46 OUTPUT
PRINT @R46
PRINT @p46
```

```
DECLARE @R47 varchar(255)
DECLARE @p47 float
EXEC altaDescarrega '2/2/2013','APL2', 'IOS', '1', 5, 5, 1, 0, @preu=@p47 OUTPUT,
@RSP=@R47 OUTPUT
PRINT @R47
PRINT @p47
```

```
DECLARE @R48 varchar(255)
DECLARE @p48 float
EXEC altaDescarrega '1/1/2014','APL2', 'IOS', '1', 5, 5, 1, 0, @preu=@p48 OUTPUT,
@RSP=@R48 OUTPUT
PRINT @R48
PRINT @p48
```

```
DECLARE @R49 varchar(255)
DECLARE @p49 float
EXEC altaDescarrega '1/4/2014','APL2', 'IOS', '1', 5, 5, 1, 0, @preu=@p49 OUTPUT,
@RSP=@R49 OUTPUT
PRINT @R49
PRINT @p49
```

```
DECLARE @R50 varchar(255)
DECLARE @p50 float
EXEC altaDescarrega '1/4/2014','APL3', 'IOS', '1', 5, 5, 1, 0, @preu=@p50 OUTPUT,
@RSP=@R50 OUTPUT
PRINT @R50
PRINT @p50
```

```
DECLARE @R51 varchar(255)
DECLARE @p51 float
EXEC altaDescarrega '1/4/2015','APL3', 'IOS', '1', 5, 5, 1, 0, @preu=@p51 OUTPUT,
@RSP=@R51 OUTPUT
PRINT @R51
PRINT @p51
```

```
DECLARE @R52 varchar(255)
DECLARE @p52 float
EXEC altaDescarrega '1/4/2015','APL3', 'IOS', '2', 5, 5, 1, 0, @preu=@p52 OUTPUT,
@RSP=@R52 OUTPUT
PRINT @R52
PRINT @p52
```

```
DECLARE @R53 varchar(255)
DECLARE @p53 float
EXEC altaDescarrega '20/1/2013','APL1', 'AND', '1', 6, 6, 1, 0, @preu=@p53 OUTPUT,
@RSP=@R53 OUTPUT
PRINT @R53
PRINT @p53
```

```
DECLARE @R54 varchar(255)
DECLARE @p54 float
EXEC altaDescarrega '20/1/2013','APL1', 'AND', '1', 7, 7, 1, 0, @preu=@p54 OUTPUT,
@RSP=@R54 OUTPUT
PRINT @R54
PRINT @p54
```

```
DECLARE @R55 varchar(255)
DECLARE @p55 float
EXEC altaDescarrega '20/1/2013','APL1', 'AND', '1', 8, 8, 1, 0, @preu=@p55 OUTPUT,
@RSP=@R55 OUTPUT
PRINT @R55
PRINT @p55
```

```
DECLARE @R56 varchar(255)
DECLARE @p56 float
EXEC altaDescarrega '20/1/2013','APL1', 'AND', '1', 8, 9, 1, 0, @preu=@p56 OUTPUT,
@RSP=@R56 OUTPUT
PRINT @R56
PRINT @p56
```

```
DECLARE @R57 varchar(255)
DECLARE @p57 float
EXEC altaDescarrega '20/1/2013','APL1', 'AND', '1', 8, 10, 1, 0, @preu=@p57 OUTPUT,
@RSP=@R57 OUTPUT
PRINT @R57
PRINT @p57
```

```
DECLARE @R58 varchar(255)
DECLARE @p58 float
EXEC altaDescarrega '20/1/2013','APL2', 'AND', '1', 9, 11, 1, 0, @preu=@p58 OUTPUT,
@RSP=@R58 OUTPUT
PRINT @R58
PRINT @p58
```

```
DECLARE @R59 varchar(255)
DECLARE @p59 float
EXEC altaDescarrega '20/1/2013','APL2', 'AND', '1', 9, 12, 1, 0, @preu=@p59 OUTPUT,
@RSP=@R59 OUTPUT
PRINT @R59
PRINT @p59
```

```
DECLARE @R60 varchar(255)
DECLARE @p60 float
EXEC altaDescarrega '20/1/2013','APL2', 'AND', '1', 10, 13, 1, 0, @preu=@p60 OUTPUT,
@RSP=@R60 OUTPUT
PRINT @R60
PRINT @p60
```

```
DECLARE @R61 varchar(255)
DECLARE @p61 float
EXEC altaDescarrega '25/1/2013','APL2', 'AND', '1', 10, 13, 1, 0, @preu=@p61 OUTPUT,
@RSP=@R61 OUTPUT
PRINT @R61
PRINT @p61
```