

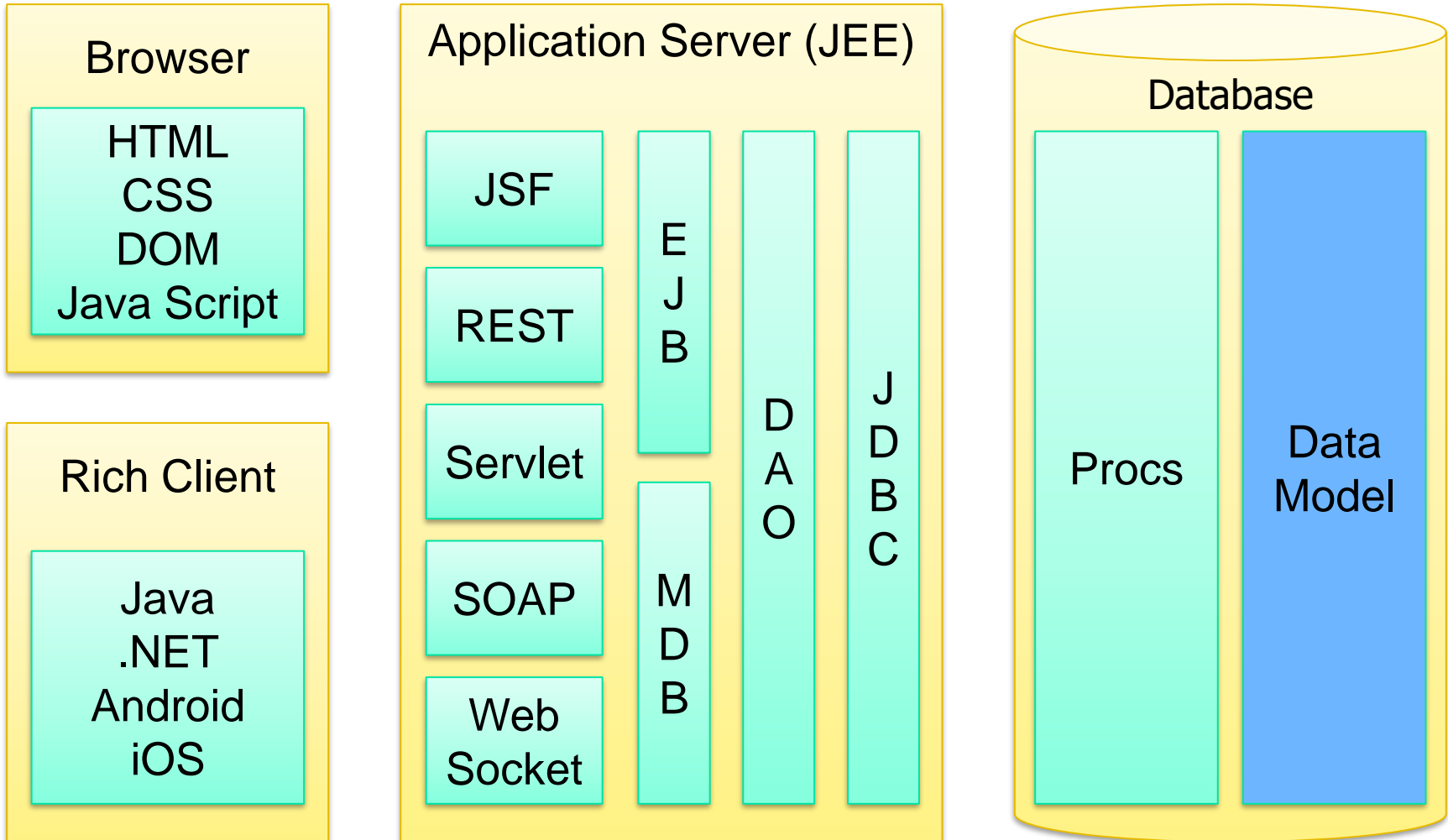


# Aplicaciones Web (parte 2)

---

Eduardo Ostertag Jenkins, Ph.D.  
OBCOM INGENIERIA S.A. (Chile)  
Eduardo.Ostertag@obcom.cl

# Modelo de datos



# PK auto generadas (1)

---

- Llave primaria (PK) **auto-generada**
  - id BIGINT AUTO\_INCREMENT NOT NULL (MySQL)
  - id NUMBER(19) NOT NULL + SEQUENCE (Oracle)
  - id BIGSERIAL NOT NULL (PostgreSQL)
  - id BIGINT IDENTITY(1,1) NOT NULL (MS-SQL)
- Llave primaria natural pasa a **UNIQUE INDEX**
  - Ejemplo típico de llave primaria natural: **RUT** chileno

# PK auto generadas (2)

## ■ MySQL

- INSERT INTO (...) VALUES (...);
- SET **\_ID** = **LAST\_INSERT\_ID()**;

## ■ Oracle

- INSERT INTO (...) VALUES (...) **RETURNING** ID INTO **ID\$**;

## ■ PostgreSQL

- **WITH** inserted **AS** (INSERT ... VALUES (...) **RETURNING** id);
- SELECT id INTO **\_id** FROM inserted;

## ■ MS-SQL

- INSERT INTO (...) VALUES (...);
- SET **@ID** = **SCOPE\_IDENTITY()**;

# Tipo de dato DATETIME

---

- Usar **DATETIME** para fechas/horas
  - Precisión mínima: **milisegundos**
  - Debe incluir la **zona horaria**
  - col DATETIME(3) DEFAULT NOW(3) (MySQL)
  - col TIMESTAMP(3) DEFAULT SYSTIMESTAMP (Oracle)
  - col TIMESTAMP(3) DEFAULT NOW() (PostgreSQL)
  - col DATETIME DEFAULT GETDATE() (MS-SQL)
- `yyyy-MM-dd 'T' HH:mm:ss.SSSXXX`
  - 2020-03-15T20:18:56.235-03:00

# Tipo de dato DECIMAL

- Usar **DECIMAL(p,s)** para guardar dinero
  - No usar: **REAL, FLOAT, SINGLE, DOUBLE**
  - En **Java** se usa el tipo de dato **BigDecimal**
  - ¿Qué valores usaría para **p** y **s** en DECIMAL?
  - Recordar: **1 peso = 0,0000350409979 UF**

- SINGLE tiene precisión de 6 a 9 dígitos
- Si sumamos número con magnitudes muy distintas podemos **perder** dígitos
- Explíquelo a un contador todo OK 😞

$$\begin{array}{r} 34.538,21 \\ + \quad \quad 1,329978 \\ \hline 34.539,539978 \end{array}$$

# Columnas NOT NULL

---

- Declarar las columnas **NOT NULL**
- NULL con cualquier cosa es NULL ☹️
- Columnas NOT NULL necesitan **DEFAULT**
- Es útil que tipos binarios (**VARBINARY**) sean NULL cuando uno quiere diferenciar entre “array de bytes de largo cero” y “null”

- Usar **UNICODE** para texto internacional
  - VARCHAR(n) → NVARCHAR(n)
  - CHAR(n) → NCHAR(n)
  - TEXT → NTEXT
  - CLOB → NCLOB
  - VARCHAR(max) → NVARCHAR(max)
  - Encoding **utf8** (PostgreSQL) o **utf8mb4** (MySQL)
  - En UNICODE: 1 carácter consume 2 bytes (fijo)
  - En UTF-8: 1 carácter consume 1, 2, 3 o 4 bytes
  - UTF-8 es igual a UNICODE, pero compacto



# ISO-8859-1 (Latin 1) (1 Byte)

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	<u>NUL</u> 0000	<u>STX</u> 0001	<u>SOT</u> 0002	<u>ETX</u> 0003	<u>EOT</u> 0004	<u>ENQ</u> 0005	<u>ACK</u> 0006	<u>BEL</u> 0007	<u>BS</u> 0008	<u>HT</u> 0009	<u>LF</u> 000A	<u>VT</u> 000B	<u>FF</u> 000C	<u>CR</u> 000D	<u>SO</u> 000E	<u>SI</u> 000F
10	<u>DLE</u> 0010	<u>DC1</u> 0011	<u>DC2</u> 0012	<u>DC3</u> 0013	<u>DC4</u> 0014	<u>NAK</u> 0015	<u>SYN</u> 0016	<u>ETB</u> 0017	<u>CAN</u> 0018	<u>EM</u> 0019	<u>SUB</u> 001A	<u>ESC</u> 001B	<u>FS</u> 001C	<u>GS</u> 001D	<u>RS</u> 001E	<u>US</u> 001F
20	<u>SP</u> 0020	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	<u>DEL</u>
80																
90																
A0	<u>NBSP</u> 00A0	ı 00A1	ı̇ 00A2	£ 00A3	* 00A4	¥ 00A5	ı̇ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	– 00AD	® 00AE	— 00AF
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C0	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
D0	Ð 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ý 00DD	Þ 00DE	ß 00DF
E0	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF
F0	ø 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ý 00FD	þ 00FE	ÿ 00FF

# Unicode (Plano 0) (2 Bytes)

- **Black** = Latin scripts and symbols
- **Light Blue** = Linguistic scripts
- **Blue** = Other European scripts
- **Orange** = Middle Eastern and SW Asian scripts
- **Light Orange** = African scripts
- **Green** = South Asian scripts
- **Purple** = Southeast Asian scripts
- **Red** = East Asian scripts
- **Light Red** = Unified CJK Han
- **Yellow** = Canadian Aboriginal scripts
- **Magenta** = Symbols
- **Dark Grey** = Diacritics
- **Light Grey** = UTF-16 surrogates and private use
- **Cyan** = Miscellaneous characters
- **White** = Unused

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Roadmap of Unicode Basic Multilingual Plane. Each numbered box represents 256 codepoints.

# ¿Cómo es la letra "ñ" en UTF-8?

- La letra "ñ" se representa con 2 bytes
  - Código de la letra "ñ" en UNICODE es F1
  - F1 = 241 (decimal) = 11110001 (binario)
  - Byte 1 de "ñ" = 11000011 = C3
  - Byte 2 de "ñ" = 10110001 = B1
- Mostrar los bytes de la letra "ñ" con **Notepad++** y **Visual Studio**

# ¿Cómo funciona su lenguaje favorito?

Lenguaje	Tipo	Representación	Bytes por carácter
Visual Basic 6	String	Unicode	2
Java	String	Unicode	2
.NET	String	Unicode	2
JavaScript	String	Unicode	2
ActionScript	String	Unicode	2
T-SQL (MS-SQL)	Varchar, Char, Text	Se especifica	1
PL/SQL (Oracle)	Varchar, Char, Clob	Se especifica	1
T-SQL (MS-SQL)	Nvarchar, Nchar, Ntext	Unicode	2
PL/SQL (Oracle)	Nvarchar, Nchar, Nclob	Unicode	2

No use tipos VARCHAR, CHAR, CLOB y TEXT para texto internacional. Use NVARCHAR, NCHAR, NCLOB, NTEXT



OBCOM

Muchas gracias

---

Muchas

gracias