



**UNIVERSIDAD SIMÓN BOLÍVAR  
DECANATO DE ESTUDIOS PROFESIONALES  
COORDINACIÓN DE INGENIERÍA GEOFÍSICA**

**CARACTERIZACIÓN DE YACIMIENTO USANDO CLASIFICACIÓN BAYESIANA  
DE ATRIBUTOS SÍSMICOS, CAMPO BLACKFOOT, CANADÁ**

Por:  
**JULIO CÉSAR MÁRQUEZ MENDOZA**

**PROYECTO DE GRADO**  
Presentado ante la Ilustre Universidad Simón Bolívar  
como requisito parcial para optar al título de  
Ingeniero Geofísico

**Sartenejas, 5 de Diciembre de 2014**



**UNIVERSIDAD SIMÓN BOLÍVAR  
DECANATO DE ESTUDIOS PROFESIONALES  
COORDINACIÓN DE INGENIERÍA GEOFÍSICA**

**CARACTERIZACIÓN DE YACIMIENTO USANDO CLASIFICACIÓN BAYESIANA  
DE ATRIBUTOS SÍSMICOS, CAMPO BLACKFOOT, CANADÁ**

Por:  
Julio César Marquez Mendoza

Realizado con la asesoría de:  
Dra. Milagrosa Aldana

**PROYECTO DE GRADO**  
Presentado ante la Ilustre Universidad Simón Bolívar  
como requisito parcial para optar al título de  
Ingeniero Geofísico

**Sartenejas, 5 de Diciembre de 2014**



**UNIVERSIDAD SIMÓN BOLÍVAR**  
**DECANATO DE ESTUDIOS PROFESIONALES**  
**COORDINACIÓN DE INGENIERÍA GEOFÍSICA**

**ACTA DE EVALUACIÓN DEL PROYECTO DE GRADO**

CÓDIGO DE LA ASIGNATURA: EP \_\_\_\_\_ FECHA: \_\_/\_\_/\_\_\_\_

ESTUDIANTE: \_\_\_\_\_ CARNET: \_\_\_\_\_

TÍTULO DEL TRABAJO: \_\_\_\_\_

\_\_\_\_\_

TUTOR: Prof. \_\_\_\_\_ CO-TUTOR: Prof. \_\_\_\_\_

JURADO: Profs. \_\_\_\_\_

APROBADO:  REPROBADO:

OBSERVACIONES:

\_\_\_\_\_

\_\_\_\_\_

El Jurado considera **por unanimidad** que el trabajo es EXCEPCIONALMENTE BUENO:

SI:  NO:  En caso positivo, justificar razonadamente: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_  
**Jurado**

\_\_\_\_\_  
**Jurado**

\_\_\_\_\_  
**Tutor Académico**

\_\_\_\_\_  
**Co-Tutor**

\_\_\_\_\_  
**Jurado**

**Nota: Colocar los sellos de los respectivos Departamentos. Para jurados externos, usar sello de la Coordinación**

## RESUMEN

En este proyecto se aplicaron los conceptos y definiciones básicas asociadas a la interpretación de registros de pozos, la teoría de decisión bayesiana y a generación e interpretación de atributos sísmicos en el Campo Blackfoot, Alberta, Canadá. La data fue adquirida en 1995 y reprocesada en 1998. Los datos sísmicos y los registros de pozos pertenecen a la Universidad de Calgary, específicamente al grupo de investigación CREWES perteneciente al Departamento de Geología y Geofísica. Para la consecución de los objetivos, fue automatizado y optimizado en su mayoría la metodología realizada, con la generación de códigos computacionales en lenguaje Python para el trabajo con registros de pozos y de atributos sísmicos y el uso del Software Petrel para la generación de los atributos necesarios. Se identificaron cuerpos arenosos en la parte noreste y un posible canal de arenas en la parte superior del campo, teniendo un contenido de arena en promedio del 70%. Se detectó agua y gas en la mitad de los pozos, pero ningún fluido pudo ser clasificado correctamente utilizando las PDF (Funciones de densidad de Probabilidad). Los registros de pozos, Sónico P, Densidad Volumétrica e Impedancia Acústica pudieron discriminar en no menos de un 70% las arenas presentes, mientras que la combinación de ellos en gráficos bivariados clasificaron las arenas con un rango de confiabilidad del 80-99%. En cuanto a la información sísmica, los atributos en cada pozos discriminaron muy bien las litologías presentes, al igual que sus combinaciones bivariadas correspondientes. Pero, al tomar en cuenta la contribución de cada pozos en PDFs promedios, ningún atributo pudo realizar ninguna clasificación confiables. Además. De los 55 combinaciones bivariadas de los atributos, sólo cuatro de ellas generaron buenos porcentajes de clasificación de más del 80%. Por último, la clasificación litológica de la sísmica a través de los registros de pozo no fue posible, se evidenciaba diferencias entre el 20% y 50% con respecto a la clasificación litológica realizada con los registros de pozos.



# DEDICATORIA

A mi madre,

A la familia,

A los pana,

A Dios.

## **AGRADECIMIENTOS**

A todos aquellos que me ayudaron en la tesis.

# ÍNDICE GENERAL

<b>RESUMEN .....</b>	<b>iv</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>xii</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>xvii</b>
<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1 MARCO GEOGRÁFICO Y GEOLÓGICO .....</b>	<b>4</b>
1.1 Localización del área de estudio.....	4
1.2 Geología regional .....	5
1.3 Geología local.....	7
<b>CAPÍTULO 2 MARCO TEÓRICO.....</b>	<b>10</b>
2.1 Teorema de Bayes .....	10
2.2 Funciones de densidad de probabilidad.....	11
2.3 Registros de Pozos.....	14
2.3.1 Registro de Rayos Gamma (Gamma Ray) .....	15
2.3.2 Volumen de arcilla.....	17
2.3.3 Registro Sónico.....	18
2.3.4 Porosidad .....	19
2.3.4.1 Porosidad Neutrón .....	20
2.3.4.2 Porosidad del registro de densidad .....	20
2.3.4.3 Porosidad aparente.....	21
2.3.5 Saturación de Agua.....	21

2.4 Propiedades derivadas de los registros de pozos .....	22
2.4.1 Impedancia acústica y elástica.....	22
2.4.2 Módulos elásticos .....	23
2.4.2.1 Módulo de Cizalla .....	23
2.4.2.2 Constante de Lamé .....	23
2.4.2.3 Relación de Poisson.....	24
2.4.2.4 Módulo de Young.....	24
2.5 Modelo convolucional de la traza sísmica.....	25
2.6 Resolución vertical sísmica .....	27
2.7 Atributos sísmicos .....	27
<b>CAPÍTULO 3 MARCO METODOLÓGICO .....</b>	<b>29</b>
3.1 Carga de datos sísmicos.....	31
3.2 Carga de los registros de pozos .....	32
3.3 Cálculos de coordenadas a una cierta profundidad en pozos desviados .....	35
3.4 Interpretación petrofísica.....	37
3.5 Cálculos de probabilidades a priori .....	38
3.6 Generación de las Funciones de densidad de probabilidad .....	40
3.7 Generación de la matriz de decisión Bayesiana .....	41
3.8 Generación de mapas de probabilidad usando los atributos sísmicos.....	43
<b>CAPÍTULO 4 RESULTADOS Y ANÁLISIS .....</b>	<b>45</b>
4.1 Coordenadas de los pozos en la zona de estudio .....	45
4.2 Control de calidad de los registros de pozos .....	46

4.3 Identificación de cuerpos arenosos y lutíticos.....	49
4.4 Contenido de arena (Probabilidad a Priori) .....	53
4.5 Discriminación litológica .....	54
4.5.1 Curvas PDFs Univariadas.....	55
4.5.1.1 Sónico P.....	55
4.5.1.2 Impedancia acústica.....	60
4.5.1.3 Densidad volumétrica.....	65
4.5.2 Curvas PDFs bivariadas.....	70
4.6 Discriminación de fluidos.....	73
4.7 Discriminación de litología con atributos sísmicos.....	75
4.7.1 Curvas PDFs univariadas.....	76
4.7.2 Curvas PDFs bivariadas.....	77
4.7.3 Clasificación litológica de atributos sísmicos .....	79
4.7.3.1 Curvas PDFs univariadas promedios.....	79
4.7.3.2 Curvas PDFs bivariadas promedios.....	83
4.7.3.3 Mapa de clasificación litológica derivados de las Curvas PDFs bivariadas promedios .....	86
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>89</b>
<b>BIBLIOGRAFÍA .....</b>	<b>92</b>
<b>APÉNDICE .....</b>	<b>95</b>
<b>Apéndice A. Tablas asociadas a matrices de decisión bayesiana para la discriminación litológica de las PDF univariadas.....</b>	<b>95</b>

<b>Apéndice B. Tablas asociadas a matrices de decisión bayesiana para la discriminación litológica de las PDFs bivariadas.....</b>	<b>99</b>
<b>Apéndice C. Código para hallar las coordenadas de los pozos en la zona de interés.....</b>	<b>102</b>
<b>Apéndice D. Código para realizar la descomposición espectral .....</b>	<b>105</b>
<b>Apéndice E. Códigos para el manejo de registros de pozos.....</b>	<b>106</b>
<b>Apéndice F. Cuaderno IPython de ejemplo para la interpretación petrofísica.....</b>	<b>132</b>

## ÍNDICE DE FIGURAS

Figura 1.1: Ubicación del Campo Blackfoot. El recuadro amarillo representa la ubicación del levantamiento sísmico en la zona de interés. (Modificado de (Bautista , 2011)).....	4
Figura 1.2: Modelo de cuenca antepaís del Jurásico Tardío en la zona de estudio. (Modificado de (Price, 1994)).....	5
Figura 1.3: Modelo de cuenca antepaís del Eoceno Temprano en la zona de estudio. (Modificado de (Price, 1994)).....	6
Figura 1.4: Formación de cuenca antepaís como consecuencia del basculamiento en el Jurásico medio (Modificado de (Price, 1994)).....	7
Figura 1.5: Columna estratigráfica de rocas Cretácicas en el área del Campo Blackfoot. (Modificado de (Bautista , 2011)).....	8
Figura 2.1: Ajuste de una curva suavizada que representa la tendencia de un histograma. Tomado de (Kay, 2014)......	12
Figura 2.2: Forma general de una curva normal con una media de 10 y una desviación estándar de 2 unidades. Modificado de (Cyders, 2012).....	13
Figura 2.3: Representación gráfica del área bajo una curva PDF para un rango de valores entre $a$ y $b$ . .....	14
Figura 2.4: Efectos de los diferentes tipo de litología en el registro de Rayos Gamma (Tomado de (Glover, 2014)).....	16
Figura 2.5: Efectos de los diferentes tipo de litología en el registro Sónico (Tomado de (Glover, 2014)).....	19
Figura 2.6: Esquema del modelo convolucional de la traza sísmica (Tomado de (Lacedre Ricciardi, 2014)). .....	26

Figura 3.1: Representación esquemática de las etapas principales involucradas en este trabajo de investigación.....	30
Figura 3.2: Posición relativa de inlines y crosslines del patch Glauconítico en el Campo Blackfoot (rectángulo negro y resaltado de la izquierda). Tomado de (Lu & Margrave, 1998).....	31
Figura 3.3: Ubicación de las bases de los pozos en el Campo Blackfoot.....	33
Figura 3.4: Representación espacial de los pozos desviados, intersectados por la profundidad media de la zona de interés (rectángulo verde) .....	36
Figura 3.5: Representación de un pozo desviado u intersectado por un plano. ....	36
Figura 3.6: Espesores de un grupo litológico (zona sombreada) para el cálculo de su probabilidad a priori. ....	39
Figura 3.7: <b>a)</b> Histogramas asociados a dos grupos pertenecientes a un propiedad. <b>b)</b> Funciones de densidad de probabilidad que representan el comportamiento de los histogramas en a). <b>c)</b> Funciones de densidad de probabilidad de a) multiplicadas por la probabilidad a priori de cada grupo. ....	40
Figura 3.8: Gráficos bivariados que representan la distribución estadísticas de dos litologías para dos propiedades distintas. ....	41
Figura 3.9: Curvas probabilísticas con para la generación de la matrices de decisión bayesiana. a) Curvas de probabilidad asociadas a una propiedad. b) Curvas de probabilidad cuyos valores son iguales a cero en aquellas zonas que estuviesen por debajo de la curva restante.....	42
Figura 3.10: Representación de la extracción de valores de amplitud a cierta profundidad correspondientes a dos trazas de dos atributos. ....	44
Figura 4.1: A la izquierda la representación de las coordenadas de los topes de cada pozo y a la derecha las coordenadas de los pozos en la zona de estudio. ....	46
Figura 4.2: Registro de Rayos Gamma perteneciente al pozo 16-08. EL registro de la izquierda representa la data original sin ninguna modificación, mientras que el de la derecha fue mejorado con un filtro mediano con una ventana de 11muestras.....	48

Figura 4.3: Registro de Volumen de arcilla perteneciente al pozo 09-05. El registro de la izquierda representa la data original sin ninguna modificación, mientras que el de la derecha fue mejorado con un filtro mediano con una ventana de 11 muestras.....	49
Figura 4.4: Registros de volúmenes de arcilla. a) Registros pertenecientes al pozo 09-17 b) Registros perteneciente al pozo 01-17.....	50
Figura 4.5: a) Registros de volumen de arcilla del pozos 01-08. b) Registro de volumen de arcilla del pozo 13-16. c) Registro de volumen de arcilla del pozo 05-16.....	51
Figura 4.6: Interpretación de cuerpos de arena a nivel en la formación Glauconítica a) Cuerpos arenosos identificados con la interpretación de los registros de Rayos Gamma b) Interpretación realizada por (Yang, et al., 1996). .....	52
Figura 4.7: Probabilidad a priori (o contenido) de arenas, donde el valor máximo representa 100% de arena y 0% de lutita. ....	54
Figura 4.8: Curvas PDFs asociadas al registro sísmico P para los pozos 01-08, 01-17, 05-16 y 08-08. ....	56
Figura 4.9: Curvas PDFs asociadas al registro sísmico P para los pozos 09-05, 09-17, 11-08 y 12-16. ....	57
Figura 4.10: Curvas PDFs asociadas al registro sísmico P para los pozos 13-16, 14-09 y 16-08. .	58
Figura 4.11: Arena clasificada como arena en el registro Sísmico P. En el gráfico de la izquierda la probabilidad litológica a priori no está incluida pero sí en el gráfico de la derecha. ....	60
Figura 4.12: Curvas PDFs asociadas al registro de impedancia acústica para los pozos 01-08, 01-17, 05-16 y 08-08. ....	61
Figura 4.13: Curvas PDFs asociadas al registro de impedancia acústica para los pozos 09-05, 09-17, 11-08 y 12-16. ....	62
Figura 4.14: Curvas PDFs asociadas al registro de impedancia acústica para los pozos 13-16, 14-09 y 16-08.....	63

Figura 4.15: Arena clasificada como arena en el registro de impedancia acústica. En el gráfico de la izquierda la probabilidad litológica a priori no está incluida pero sí en el gráfico de la derecha. ....	65
Figura 4.16: Curvas PDFs asociadas al registro de densidad volumétrica para los pozos 01-08, 01-17, 05-16 y 08-08. ....	66
Figura 4.17: Curvas PDFs asociadas al registro de densidad volumétrica para los pozos 09-05, 09-17, 11-08 y 12-16. ....	67
Figura 4.18: Curvas PDFs asociadas al registro de densidad volumétrica para los pozos 13-16, 14-09 y 16-08. ....	68
Figura 4.19: Arena clasificada como arena en el registro de densidad bulk. En el gráfico de la izquierda la probabilidad litológica a priori no está incluida pero sí en el gráfico de la derecha. ....	70
Figura 4.20: Representación gráfica de las Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Sónico P e Impedancia acústica. ....	71
Figura 4.21: Representación gráfica de las Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Sónico P y Densidad Volumétrica. ....	72
Figura 4.22: Representación gráfica de las Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Impedancia Acústica y Densidad Volumétrica. ....	73
Figura 4.23: Curvas PDFs asociadas al registro de densidad del pozo 01-08. ....	75
Figura 4.24: Curvas PDFs asociadas al registro de densidad del pozo 01-08. ....	75
Figura 4.25: Curvas PDF pertenecientes al atributo de la Segunda Deriva del pozo 01-08. ....	77
Figura 4.26: Curvas PDF promedios asociadas a cuatro atributos sísmicos distintos: Impedancia acústica Relativa, Coseno de la fase, Frecuencia Dominante y Primera Derivada. ....	80
Figura 4.27: Curvas PDF promedios asociadas a cuatro atributos sísmicos distintos: Frecuencia instantánea, Fase instantánea, Amplitud original y Desfase en fase. ....	81
Figura 4.28: Curvas PDF promedios asociadas a cuatro atributos sísmicos distintos: Segunda derivada, Amplitud RMS, Amplitud original y Caos. ....	82

Figura 4.29: PDFs bivariadas para la combinación de atributos Impedancia Acústica Relativa vs Frecuencia Dominante. El grafico de la derecha tiene el aporte de la probabilidad a priori litológica. .... 83

Figura 4.30: Curvas PDFs bivariadas para la combinación..... 85

Figura 4.31: a) Mapa de probabilidad a priori (contenido de arena) derivado de la interpretación petrofísica b) Mapa de contenido de arena derivado de la clasificación por PDF bivariadas de atributos sísmicos correspondientes s la combinación Frecuencia Dominante vs. Frecuencia Instantánea..... 87

Figura 4.32: a) Mapa de probabilidad a priori (contenido de arena) derivado de la interpretación petrofísica b) Mapa de contenido de arena derivado de la clasificación por PDF bivariadas de atributos sísmicos correspondientes a la combinación Impedancia Acústica Relativa vs Primera Derivada..... 88

## ÍNDICE DE TABLAS

Tabla 3.1: Coordenadas UTM, zona 12U, de los vértices del Patch Glauconítico. ....	32
Tabla 3.2: Coordenadas UTM, zona 12U, de los pozos proporcionados .....	33
Tabla 3.3: Inventario de registros disponibles en los pozos del Campo Blackfoot.....	34
Tabla 3.4: Matriz de decisión bayesiana de ejemplo, resultante del procedimiento propuesto en este apartado. ....	43
Tabla 4.1: Porcentaje de clasificación correcto de la arena para el registro de sónico P, derivados de las matrices de decisión Bayesiana.....	59
Tabla 4.2: Matrices de decisión Bayesiana simplificada para el registro de impedancia acústica.....	64
Tabla 4.3: Matrices de decisión Bayesiana simplificada para el registro de densidad bulk.....	69
Tabla 4.4: Presencia de fluidos en los pozos del campo Blackfoot.....	74
Tabla 4.5: Discriminación litológica aceptable derivada de las PDF para los diferentes atributos sísmicos generados. Los recuadros en rojos representan una discriminación no válida. ....	77
Tabla 4.6: Porcentaje de clasificación correcto asociados a las arenas para tres tipos de combinación de atributos sísmicos.....	79
Tabla 4.7: Combinaciones de Atributos sísmicos que representaban zonificaciones correctas de los grupos litológicos en las PDF bivariadas correspondientes. ....	84
Tabla 0.1: Matrices de decisión Bayesiana simplificada para el registro de sónico P. ....	95
Tabla 0.2: Matrices de decisión Bayesiana simplificada para el registro de impedancia acústica.....	96
Tabla 0.3: Matrices de decisión Bayesiana simplificada para el registro de densidad bulk.....	96
Tabla 0.4: Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Sónico P e Impedancia acústica.....	97

Tabla 0.5: Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Sónico P y Densidad Volumétrica. ....	97
Tabla 0.6: Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Impedancia Acústica y Densidad Volumétrica.....	98
Tabla 0.7: Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Sónico P e Impedancia acústica.....	99
Tabla 0.8: Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Sónico P y Densidad Volumétrica.....	100
Tabla 0.9: Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Impedancia Acústica y Densidad Volumétrica.....	101

## INTRODUCCIÓN

En la mayoría de las ciencias existe la necesidad realizar correctas interpretaciones que involucra la toma de decisiones basadas en el análisis, usando métodos estadísticos de la información disponible (López Casuso, 2009), con el aprovechamiento de los avances computacionales que se han generado en la últimas décadas. En cuanto a las geociencias, especialmente el nexo que existe en la información petrofísica y la información sísmica, se han hecho usas de varias técnicas estadísticas que permitan evaluar la información disponible y sobre todo reducir estéticamente la incertidumbre presente en los métodos geofísicos indirectos más usados.

Las incertidumbres de exploración de yacimiento se han podido cuantificar y manejar por medio de modelos petrofísicos a través de técnicas de reconocimientos de patrones para mejorar la interpretación de la información sísmica. La petrofísica nos ayuda a establecer los enlaces entre la respuesta sísmica y las propiedades de los yacimientos, y así extender la información disponible para los análisis de los sistemas de clasificación. Además el cálculo de atributos sísmicos permite, entre otros aspectos, visualizar información geológica no evidente en el proceso de interpretación sísmica. Esta información puede obtenerse utilizando métodos de reconocimiento de patrones para evaluar los valores de los atributos obtenidos, así como en el caso de la evaluación petrofísica. Entre estos métodos se encuentra los de Clasificación Bayesiana basados en funciones de probabilidad no paramétrica, que ayudan de forma cuantitativa a medir la incertidumbre de la interpretación y la el riesgo de clasificar incorrectamente la información disponibles en cualquier ubicación espacial.

La identificación de eventos estratigráficos y estructurales a pequeñas escalas con son difíciles de identificar con solo la información sísmica debido a los límites de resolución. La evaluación petrofísica ayuda a resolver este problema y con la ayuda de modelos de clasificación estadística, se puede propagar es información en todo la sísmica con el fin de disminuir la incertidumbre que involucra el hecho de no tener estudios petrofísicos en todo el yacimiento (Avseth, Mukerji, & Mavko, 2010). Combinar los modelos físicos determinísticos con las técnicas estadísticas nos

ayudan a desarrollar nuevos métodos para la interpretación y la estimación de propiedades de las rocas del yacimiento desde la data sísmica.

Estas técnicas estadísticas son útiles y necesarias para reducir la incertidumbre en nuestras interpretaciones, pero ninguna técnica de manipulación de la información estadística es poderosa sino se tiene la experticia para el análisis y el entendimiento físico de los procesos involucrados en el yacimiento (Avseth, Mukerji, & Mavko, 2010). La mejor estrategia es la de combinar las estadísticas computacionales con un buen conocimiento humano para el análisis del yacimiento.

Este estudio pretende caracterizar un yacimiento en el campo Blackfoot en Canadá, a partir de la clasificación Bayesiana de la información sísmica y de los atributos sísmicos convencionales. Para ello se cuenta con un cubo sísmico de datos post-apilamiento y 12 pozos en el área. Se verá como cuantificar las incertidumbres en la caracterización del yacimiento y a manejar por la combinación la evaluación de propiedades petrofísicas (e.g. litología, contenido de hidrocarburos) y el reconocimiento de patrones para interpretar los atributos sísmicos, utilizando la teoría Bayesiana.

El Campo Blackfoot se localiza a 15 km al sureste del pueblo Strathmore, Alberta, Canadá. Dentro del área de estudio el grupo Mannville suprayace inconforme a los carbonatos Missisipianos y las rocas de interés corresponden a valles incisos rellenos por sedimentos dentro de la formación glauconítica, que consiste de arenas cuarzosas de grano muy fino a medio (Price, 1994). El hidrocarburo principal presente en Blackfoot es el petróleo, y es posible encontrar gas en menor proporción.

Debido a que la cantidad de pozos y a la variedad de atributos sísmicos disponibles traía en sí un trabajo arduo, se trató de optimizar y automatizar cada una de las operaciones necesarias para generar los resultados. Esto se logró tratando de crear la cantidad de código necesario para ser reusado en la mayoría de los procedimientos involucrados en la evaluación de los resultados generados en este trabajo de investigación.

En la metodología usada se trató de desligarse de la dependencia absoluta de programas computacionales con el fin de tener el control y un buen entendimiento de las tareas realizadas. Esta metodología de trabajo se basó en la expuesta por (Avseth, Mukerji, & Mavko, 2010) y se extendió para llevar a cabo el análisis discriminatorio generado con los atributos sísmicos.

# CAPÍTULO 1

## MARCO GEOGRÁFICO Y GEOLÓGICO

### 1.1 Localización del área de estudio

Los datos sísmicos utilizados corresponden al Campo Blackfoot, ubicado a 20 Km del sureste de Strathmore Alberta y a 45 km al sureste de la ciudad de Calgary, Canadá, perteneciendo a la zona UTM 12. En la Figura 1.1 se observa la disposición de los geófonos e imágenes satelitales referenciales al campo de estudio.

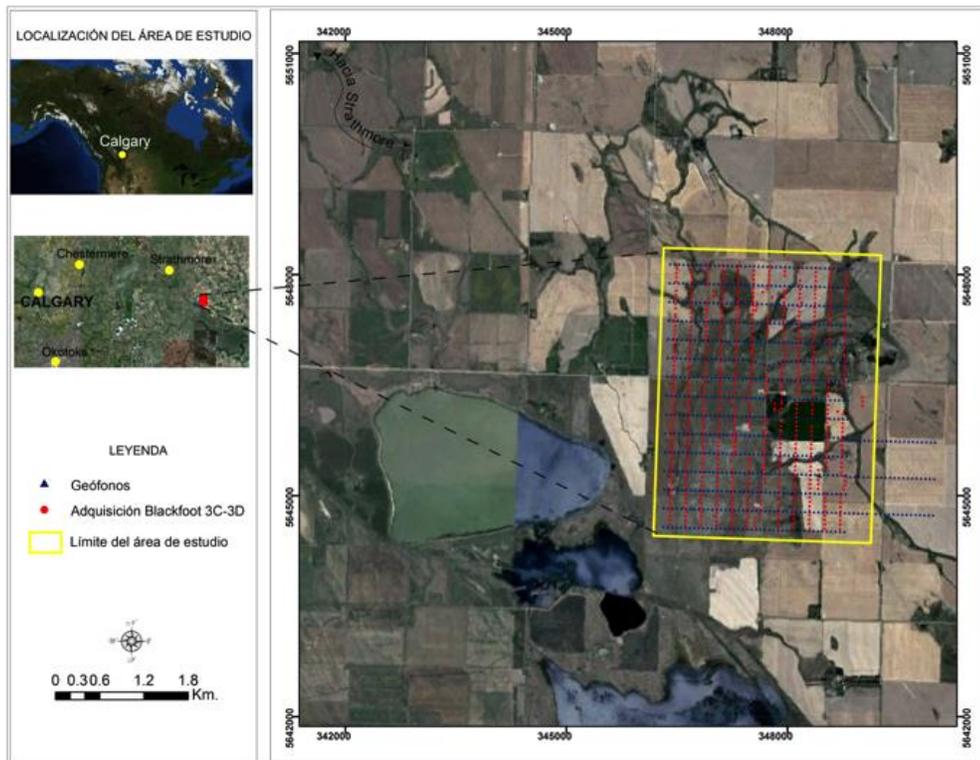


Figura 1.1: Ubicación del Campo Blackfoot. El recuadro amarillo representa la ubicación del levantamiento sísmico en la zona de interés. (Modificado de (Bautista , 2011)).

## 1.2 Geología regional

El Campo Petrolero *Blackfoot* está localizado en la Cuenca Sedimentaria del Oeste de Canadá, la cual abarca un área de aproximadamente 1.400.000km<sup>2</sup>, conformada por rocas sedimentarias, las cuales se superponen al basamento pre-cámbrico, el cual forma el Cratón de Norteamérica. Estas rocas sedimentarias se extienden desde la Cordillera Canadiense (al este), hasta el Escudo Canadiense (al oeste), cuyos espesores (de 5 a 6 kilómetros) aumentan gradualmente hacia el sur.

En una primera etapa, entre el Proterozoico superior y el Jurásico Tardío se evidencia la predominancia de rocas carbonáticas que fueron depositadas junto al margen de Norte América, en el Cratón estable Norteamericano. La mayor cantidad de sedimentos en esta etapa se depositó hacia el noreste y darían lugar posteriormente al desarrollo de la Cordillera Canadiense, que progradaría luego fuera del margen pasivo (Mossop & Shetsen, 1994). (Figura 1.2)



Figura 1.2: Modelo de cuenca antepaís del Jurásico Tardío en la zona de estudio. (Modificado de (Price, 1994))

En una segunda etapa, entre el Jurásico Tardío y el Eoceno Temprano, se evidencia la predominancia de rocas clásticas que habían sido formadas durante el margen activo del orogénico la Cordillera Canadiense y se evidencia una depositación importante de sedimentos hacia el suroeste de la zona (Mossop & Shetsen, 1994) . (Ver Figura 1.3)

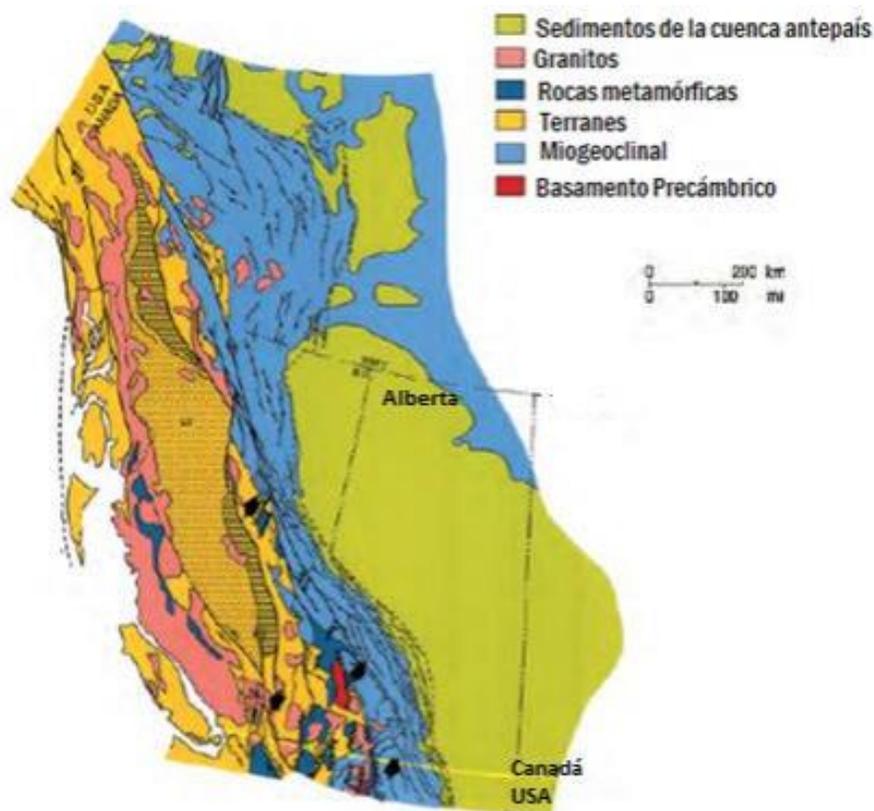


Figura 1.3: Modelo de cuenca antepaís del Eoceno Temprano en la zona de estudio. (Modificado de (Price, 1994))

Posterior a estas etapas pudieron evidenciarse patrones de inundaciones marinas, sedimentación y erosión como resultado de procesos epirogénicos, a partir de estos eventos se generarían nuevas subcuencas. La carga tectónica que se originó en el Jurásico medio como consecuencia de la colisión entre el prisma de acreción y el Cratón Norteamericano, e hicieron evidente un basculamiento de la cuenca en cuestión. (Figura 1.4).

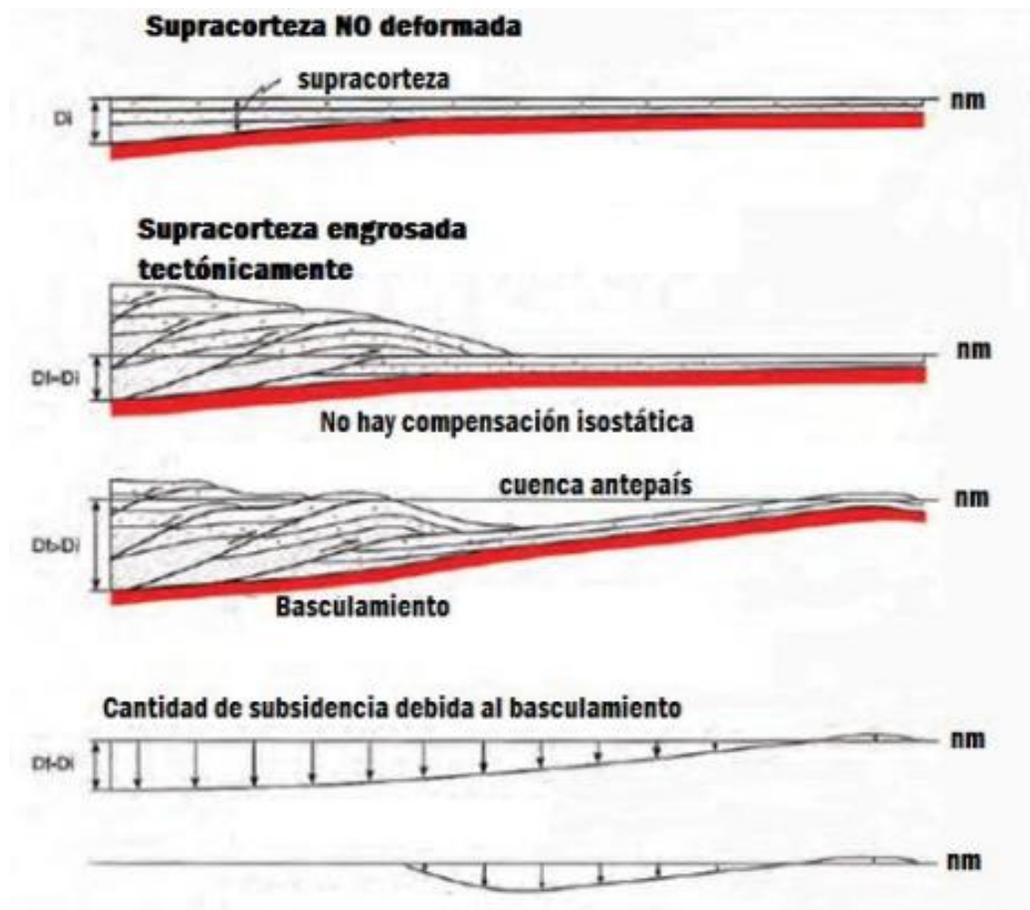


Figura 1.4: Formación de cuenca antepaís como consecuencia del basculamiento en el Jurásico medio (Modificado de (Price, 1994))

### 1.3 Geología local

En los reportes presentados se presta especial atención al Grupo Mannville y a los miembros que lo constituyen, entre ellos, el Miembro Glauconítico. Se presenta la columna estratigráfica de rocas Cretácicas del área del Campo Blackfoot (Figura 1.5).

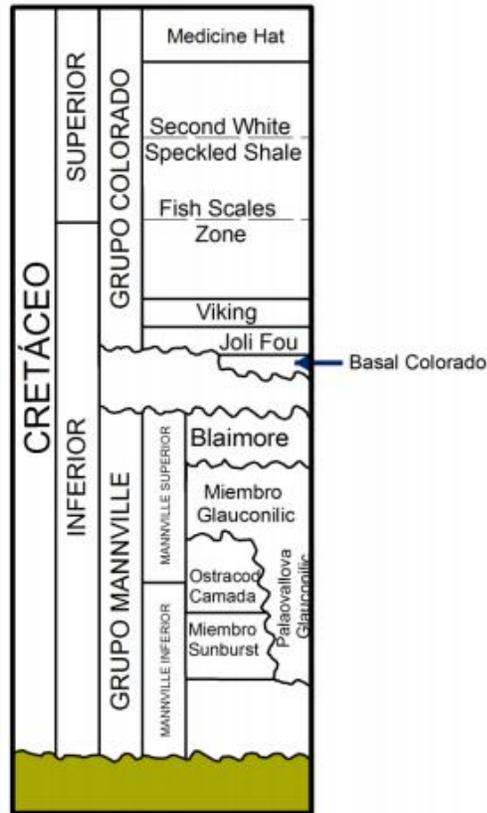


Figura 1.5: Columna estratigráfica de rocas Cretácicas en el área del Campo Blackfoot.

(Modificado de (Bautista , 2011))

El Grupo Mannville consiste de intercalaciones de arenas y lutitas no marinas que se encuentran cubiertas por un miembro fino calcáreo que a su vez se encuentra cubierto por lutitas marinas y arenas glauconíticas en la zona centro-sur de Alberta. El Grupo Mannville es suprayacente a carbonatos del Mississipiano de la Formación Shund, formando una discordancia. La superficie de contacto erosional tiene topografía irregular. El Grupo Mannville se encuentra sobre carbonatos del Paleozóico y sus sedimentos representan la etapa temprana de transgresión. Los valles incisos presentes en toda la zona de estudio describen una transgresión interrumpida por períodos (Hayes, et al., 1994).

El Grupo Mannville es responsable de una cuantiosa cantidad de reservas hidrocarburíferas en el territorio Canadiense. De él se extraen: gas, petróleo pesado y bitumen. Se estima que las reservas totales de gas en el Miembro Mannville Inferior ascienden a más de 315 millones de metros cúbicos

mientras que en Mannville Superior se tienen más de 630 millones de metros cúbicos. Las reservas de petróleo recuperable son de más de 105 millones de metros cúbicos en Mannville Inferior y de cerca de 200 millones de metros cúbicos en Mannville Superior.

El Miembro Glauconítico consiste de una arenisca silíceas de grano muy fino a medio en el área más oriental de Alberta, siendo la glauconita un material común en la zona norte-centro de Alberta. Los sedimentos en este canal se dividen en 3 unidades correspondientes a 3 fases de valles incisos. Los Miembros Inferior y Superior se componen de areniscas de cuarzo con porosidad promedio de 18%, mientras que el Miembro Medio está compuesto por una arenisca menos porosa. El hidrocarburo primario presente es el petróleo, con menor presencia de gas en el Miembro Superior. Cada miembro tiene un espesor entre 5m a 20m. (Bautista , 2011). La Formación Glauconítica es la mayor productora de hidrocarburos en el sur y centro de Alberta. La tendencia deposicional está controlada por la fluctuación del nivel del mar y la paleotopografía en la inconformidad pre-Cretácica. Adicionalmente la Formación Glauconítica se caracteriza por una secuencia progradacional deltáica y valles incisos con sedimentos fluviales y de estuarios (Hayes, et al., 1994).

## CAPÍTULO 2

### MARCO TEÓRICO

#### 2.1 Teorema de Bayes

Supongamos que hemos observado el resultado de un experimento, y preguntamos por la probabilidad de que el resultado es debido a una particular causa de una serie de posibles causas.

Sean  $B_1, B_2, \dots, B_n$ ,  $n$  sucesos mutuamente excluyentes que representan las  $n$  posibles causas del resultado del experimento, siendo  $A$  el suceso que representa el resultado. Al realizar el experimento se dio el suceso  $A$ , y se desea saber la probabilidad de que habiéndose dado el suceso  $A$ , se haya dado uno de los sucesos  $B_i$ . En otras palabras, queremos saber si la causa de que se haya dado el suceso  $A$  fue el hecho de que se dio el suceso  $B_i$ ; o sea, interesa conocer  $P(B_i / A)$ . En este caso  $P(B_i / A)$  no es la probabilidad de que se dé  $B_i$ ; o sea, interesa conocer  $P(B_i / A)$ . En este caso  $P(B_i / A)$  no es la probabilidad de que se dé  $B_i$ , dado  $A$ , sino la probabilidad de que se haya dado  $B_i$  puesto que se dio  $A$ .

$$P(B_i / A) = \frac{P(B_i \cap A)}{P(A)} \quad (2.1)$$

Sabemos que  $P(B_i \cap A) = P(A/B_i) \cdot P(B_i)$ , El suceso A debe darse en conjunción con una de las n mutuamente excluyentes causas, de forma que A ocurrirá si se da alguna de las posibles causas  $B_i$ .

Podemos escribir  $A = (A \cap B_1) \cup (A \cap B_2) \cup \dots \cup (A \cap B_n)$ , donde  $(A \cap B_i)$  y  $(A \cap B_j)$  son sucesos mutuamente excluyentes para  $i \neq j$ , luego

$$\begin{aligned}
 P(A) &= P(A \cap B_1) + P(A \cap B_2) + \dots + P(A \cap B_n) = \sum_{i=1}^n P(A \cap B_i) \\
 &= \sum_{i=1}^n P(A/B_i)P(B_i)
 \end{aligned}
 \tag{2.2}$$

Entonces:

$$P(B_i / A) = \frac{P(B_i \cap A)}{P(A)} = \frac{P(A/B_i)P(B_i)}{\sum_{i=1}^n P(A/B_i)P(B_i)}
 \tag{2.3}$$

## 2.2 Funciones de densidad de probabilidad

En términos generales, una PDF o Función de Densidad Probabilística es una versión suavizada de un histograma (Figura 2.1), es una versión más compacta y algebraica que puede ser usada fácilmente para obtener cálculos probabilísticos.

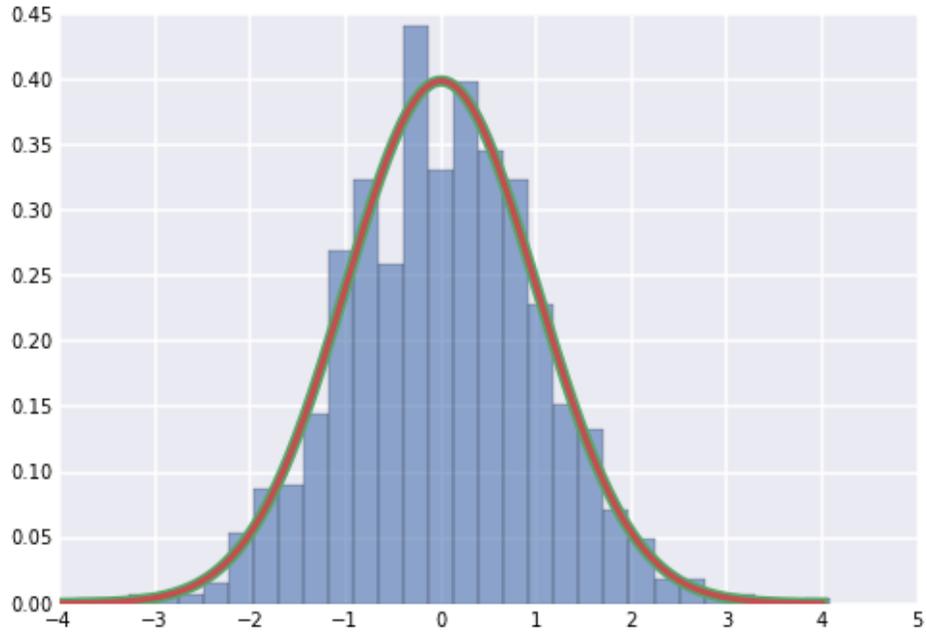


Figura 2.1: Ajuste de una curva suavizada que representa la tendencia de un histograma. Tomado de (Kay, 2014).

El tipo de PDF más popular es la curva normal o la distribución normal. Una razón para esto es que la suma de un número grande de variables aleatorias independientes usualmente se aproxima a una distribución normal que es simétrica sobre la media (ver la Figura 2.2). Estas PDFs pueden escribirse como (asumiendo una desviación estándar  $\sigma$  y una media  $x_m$ ):

$$pdf(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x - x_m)^2}{2\sigma^2}\right] \quad (2.4)$$

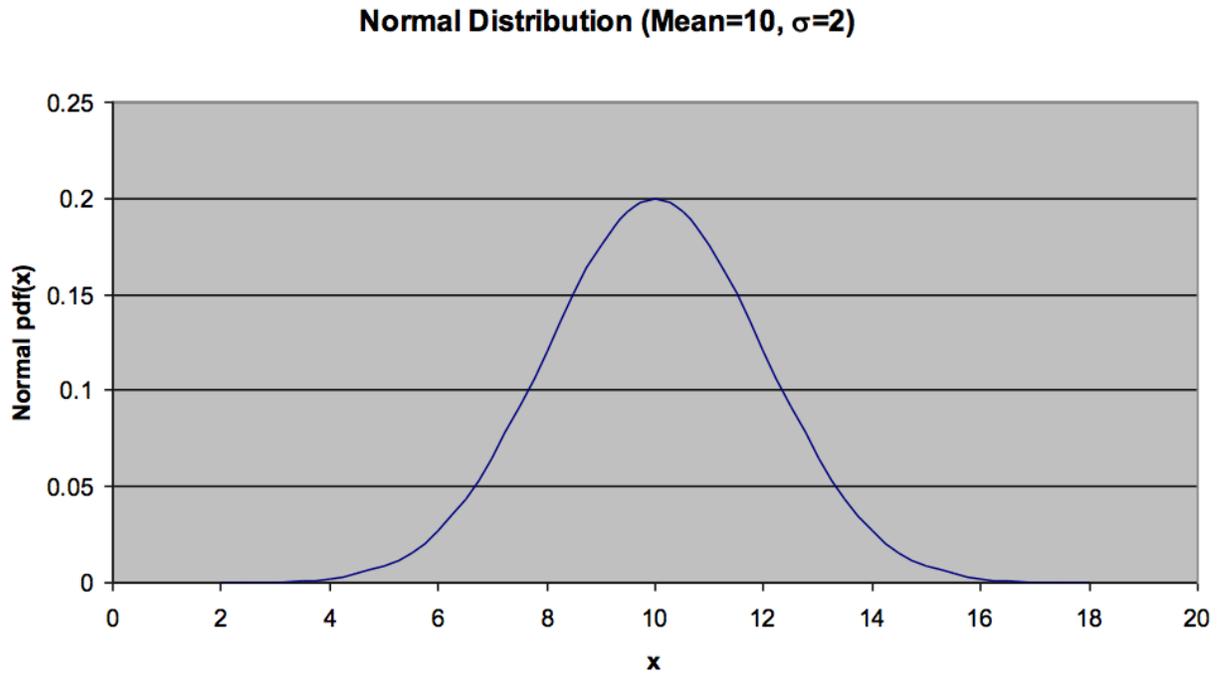


Figura 2.2: Forma general de una curva normal con una media de 10 y una desviación estándar de 2 unidades. Modificado de (Cyders, 2012)

El principal propósito del uso de las PDF es el de calcular las probabilidades de cualquier evento que pueda generarse, y existen dos rasgos muy importantes de estas funciones de probabilidad que facilitan la ejecución de este propósito:

- El área total bajo la curva (de  $-\infty$  a  $+\infty$ ) es igual a la unidad.

$$\int_{-\infty}^{\infty} pdf(x)dx = 1 \quad (2.5)$$

- La probabilidad de un rango de valores (Figura 2.3) puede ser calculada hallando el área entre los valores de una curva PDF.

$$P(a < x < b) = \int_a^b pdf(x)dx \quad (2.6)$$

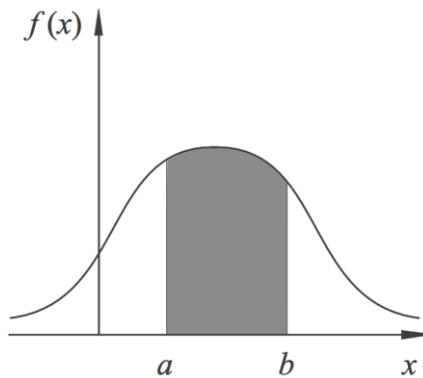


Figura 2.3: Representación gráfica del área bajo una curva PDF para un rango de valores entre  $a$  y  $b$ .

### 2.3 Registros de Pozos

Los registros de pozos representan una grabación en función de la profundidad de alguna propiedad o características asociadas a formaciones rocosas, realizadas por instrumentos de medición especiales. Este método es muy importante dentro de la exploración y producción de hidrocarburos.

### 2.3.1 Registro de Rayos Gamma (Gamma Ray)

El registro de Rayos Gamma o Gamma Ray mide la emanación de radiación total natural de una formación. Esta radiación Gamma se origina del potassium-40 y de los isótopos de Uranio-Radio y de las series de Torio (Glover, 2014).

En sedimentos, el registro refleja principalmente el contenido de lutita debido a que los minerales contienen isotopos radiactivos de cuales en la mayoría de los casos es el potasio, quien tiende a concentrarse en las arcillas y lutitas. Las cenizas volcánicas, algunos depósitos de sal y alguno sedimentos con uranio tienden a representar lecturas significativas (Sheriff, 2002), sin embargo, las lutitas tienen por mucho la radiación más fuerte considerando que no todas son necesariamente radioactivas (Rider, 1996). El principal uso del registro de rayos gamma es derivar el volumen de arcilla, pero también es usado para correlacionar, sugerir facies y secuencias que facilitan la identificación de litología y posibles ambientes depositacionales (Rider, 1996).

En la Figura 2.4 puede observarse el efecto de diferentes tipos de litología en la respuesta de los rayos gamma. La lutitas suelen presentar valores altos de radioactividad los cuales están influenciados por la presencia de arenas, lo contrario ocurre para las zonas con arenas, que presentan valores bajos de radiactividad pero no menores a las zonas con presencia de las calizas, dolomitas y carbón. También es importante resaltar que la presencia de ciertos elementos altera la respuesta radiactiva, por ejemplo, los feldespatos y micas indican valores relativamente altos de radiactividad pero elementos orgánicos en el estrato expresan en el registro de rayos gamma una respuesta significativamente muy alta.

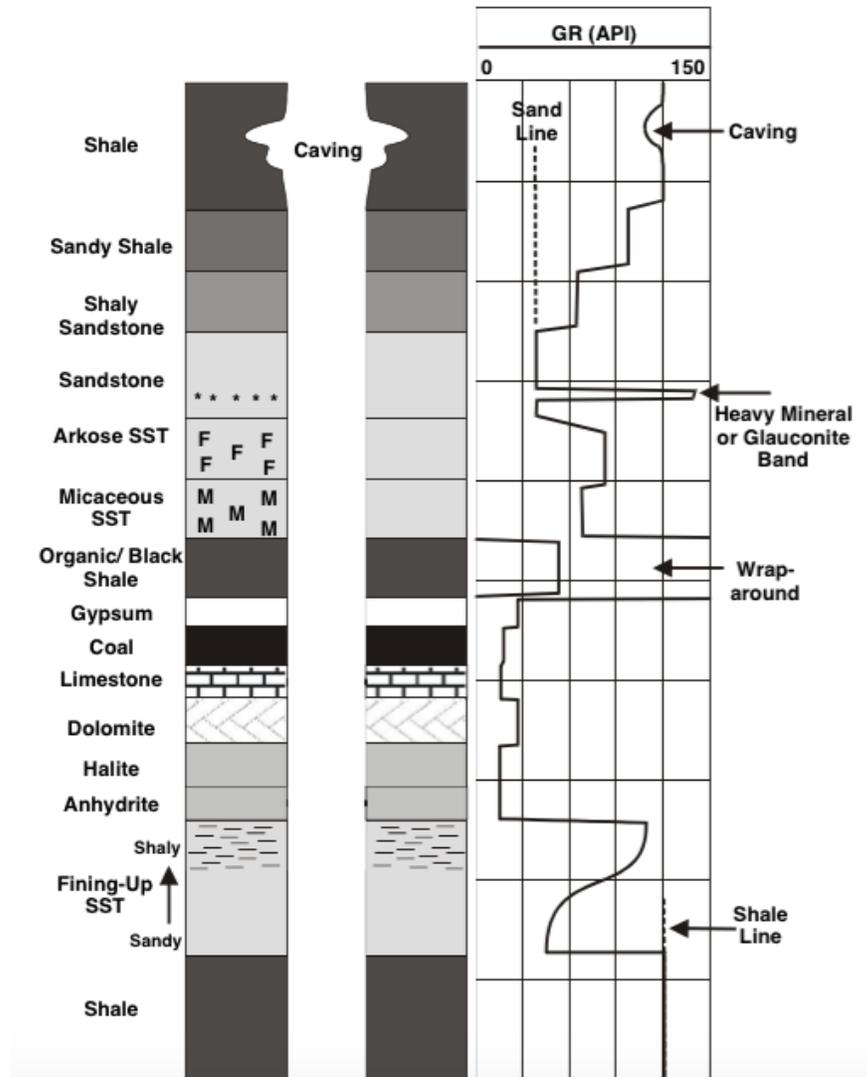


Figura 2.4: Efectos de los diferentes tipo de litología en el registro de Rayos Gamma (Tomado de (Glover, 2014))

Tomando en cuenta que las calizas tienen tamaño de grano fino y las arenas tamaño de grano un poco más grueso, podemos con la ayuda de registro de rayos gamma hablar de zonas con variaciones de tamaño de grano que pueden dar indicios de posibles ambientes sedimentarios (Falcón, 2011), tal como se observa en la parte inferior de la Figura 2.4 en el estrato que define un Fining-Up, un aumento del tamaño del grano hacia arriba.

### 2.3.2 Volumen de arcilla

Debido a que las lutitas son más radioactivas que las arenas o carbonatos, los registros de rayos gamma suelen ser usados para calcular el volumen de arcilla en un yacimiento poroso. El primer paso a realizar es el cálculo del índice de rayos gamma (o índice de Gamma Ray):

$$I_{GR} = \frac{GR_{log} - GR_{cn}}{GR_{sh} - GR_{cn}} \quad (2.7)$$

Donde:

$I_{GR}$ : Índice de Gamma Ray.

$GR_{log}$ : valor de Gamma Ray de la formación.

$GR_{cn}$ : valor mínimo de Gamma Ray asociado a la arena más limpia.

$GR_{sh}$ : valor máximo de Gamma Ray asociada a la lutita más sucia.

Mucho tiempo atrás se asumía que volumen de arcilla estaba determinado por el ya mencionado *Índice de Gamma Ray*, pero estudios posteriores establecieron relaciones no lineales entre ambas que definen de una manera un poco más acertada la cantidad de arcilla presente en la formación. Estas relaciones representan varias ecuaciones, cuyo principal criterio de escogencia es la edad de la roca. Las rocas de nuestra zona de interés pertenecen al periodo **Cretácico**, por lo tanto la ecuación de volumen de arcilla usada fue la asociada a las rocas **pre-terciarias**:

$$V_{sh} = 0.33(2^{2I_{GR}} - 1) \quad (2.8)$$

### **2.3.3 Registro Sónico**

El registro sónico representa el tiempo de tránsito de una formación, usualmente medido en microsegundos por pies. Es usado para proporcionar la información necesaria que permite soportar y calibrar la onda sísmica (Glover, 2014) y también para medir la capacidad que tiene una formación para transmitir las ondas acústicas, siendo afecta por la litología, la porosidad y la presencia de fluido en los poros (Rider, 1996).

En Figura 2.5 se puede observar la respuesta asociada al tiempo de tránsito para diferentes litologías. La respuesta sónico no está sólo afectada por el tipo litología presente sino también por el estado de compactación de la roca y la presencia de fluido. Obsérvese la diferencia de valores que puede reportar una arena compactada o una arena con presencia de un fluido, pero también observe el gran rango de valores que puede tener una lutita cuando ésta se encuentra compactada o no.

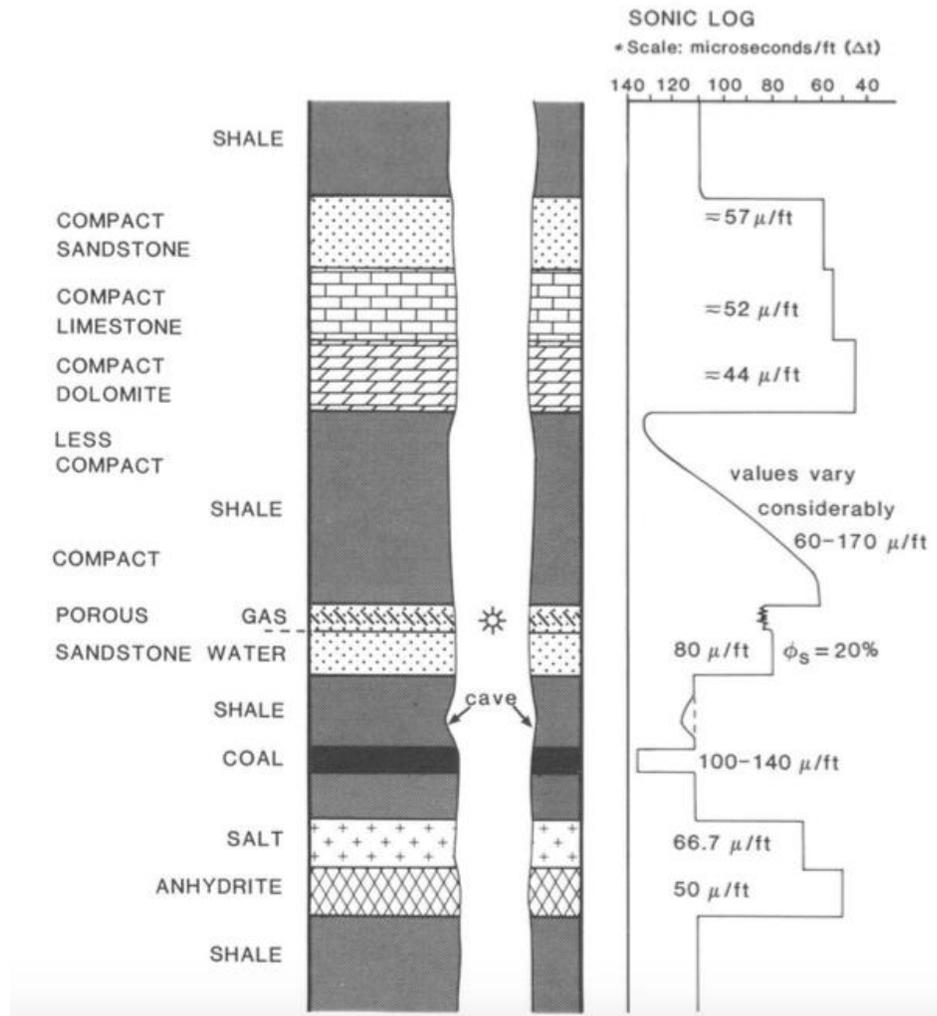


Figura 2.5: Efectos de los diferentes tipo de litología en el registro Sónico (Tomado de (Glover, 2014))

### 2.3.4 Porosidad

La porosidad de una roca puede obtenerse de los registros sónicos, registro de densidad o el registro neutrón. Para todos estos registros, la respuesta se ve afectada por la porosidad de la formación, el fluido presente y el tipo de matriz.

### 2.3.4.1 Porosidad Neutrón

Los registros de porosidad son usados principalmente para la delineación de formación porosas y la determinación de su porosidad. Ellos responden principalmente cantidad de hidrógeno presente en la formación. En una fonación limpia, una densidad de hidrogeno baja indica una porosidad de llenado de líquido baja (Schlumberger Limited, 1991).

La porosidad calculada del registro neutrón es afectada de alguna manera por la matriz de la formación y por la presencia de gas. Los valores del registro de Neutros son usados con otros registros de porosidad o análisis de núcleos para detectar la presencia de gas (Sheriff, 2002).

### 2.3.4.2 Porosidad del registro de densidad

El registro de densidad mide la densidad volumétrica (densidad bulk) de la formación. Es útil en la detección de gas y el reconocimiento de evaporitas (Glover, 2014) pero su principal uso es derivar el valor de la porosidad total de la formación. Para determinar esta porosidad  $\phi_D$ , es necesario tener una formación limpia con una densidad de matriz conocida  $\rho_m$  que contenga un fluido con una densidad promedio  $\rho_f$  y una densidad volumétrica  $\rho_b$  y hacer uso de la siguiente ecuación.

$$\phi_D = \frac{\rho_m - \rho_b}{\rho_m - \rho_f} \quad (2.9)$$

### 2.3.4.3 Porosidad aparente

La porosidad aparente toma en cuenta la contribución de la porosidad neutrón y la porosidad por densidad, definida por la ecuación (2.10), usada cuando la formación sólo tiene la presencia de agua o petróleo y la ecuación (2.11) sólo usada cuando se tiene la presencia de gas.

$$\phi_A = \frac{\phi_D + \phi_N}{2} \quad (2.10)$$

$$\phi_A = \sqrt{\frac{\phi_D + \phi_N}{2}} \quad (2.11)$$

### 2.3.5 Saturación de Agua

La saturación de agua representa la cantidad (porcentaje o fracción) de agua presente en una formación. Generalmente se asume, a menos que se pruebe lo contrario, que el volumen de poro no llenado con agua está llenado con hidrocarburos (Schlumberger Limited, 1991). Para hallar la saturación de agua se puede usar la ecuación (2.12), donde  $a$  representa el factor de cementación,  $m$  el exponente de cementación,  $\phi_A$  la porosidad aparente,  $R_t$  la resistividad de la formación y  $R_w$  la resistividad del agua de formación:

$$S_w = \sqrt[n]{\frac{aR_w}{R_t\phi_A^m}} \quad (2.12)$$

Para el cálculo de  $R_w$  se usa la ecuación (2.13), una vez identificadas las zonas con un bajo valor de volumen de arcilla y resistividad profunda (acuífero), se calcula el valor de  $R_{wA}$  el cual corresponderá al valor de la resistividad del agua de formación (Mendoza & Landa, 2005).

$$R_{wA} = \frac{R_t \phi_A^m}{a} \quad (2.13)$$

## 2.4 Propiedades derivadas de los registros de pozos

Existen propiedades de las rocas que se puede derivar de la información obtenida de los registros de pozos con el fin de extender y mejorar los análisis petrofísicos. Para este estudio sólo se usaron algunas de las más representativas.

### 2.4.1 Impedancia acústica y elástica

La impedancia acústica es la propiedad física cuyo cambio determina los coeficientes de reflexión a incidencia normal, es decir, la velocidad de la onda sísmica P multiplicada por la densidad (Sheriff, 2002):

$$Z = \rho * V_p \quad (2.14)$$

Debido a que los coeficientes de reflexión cambian con el ángulo, la impedancia acústica es a usada cuando nos referimos a situaciones de incidencia no normal (Sheriff, 2002):

$$K = \frac{1}{2} \left( \frac{V_{S_2}^2}{V_{P_2}^2} + \frac{V_{S_1}^2}{V_{P_1}^2} \right) \quad (2.15)$$

$$EI = V_P^{(1+\tan^2\theta)} V_S^{(-8K\sin^2\theta)} \rho^{(1-4K\sin^2\theta)} \quad (2.16)$$

## 2.4.2 Módulos elásticos

La elasticidad enfrenta las deformaciones que desaparecen al eliminar el esfuerzo que las genera. Existen relaciones físicas que nos permiten hallar de una manera sencilla ciertos módulos elásticos.

### 2.4.2.1 Módulo de Cizalla

También conocida como módulo de rigidez, define la relación de esfuerzo-deformación para un cizallamiento simple, es decir, define el cambio de forma que experimenta un material elástico cuando se aplican esfuerzos de cizalla. Existen ecuaciones que permiten estimar esta relación, pero la usada para este estudio está representada por la siguiente ecuación:

$$\mu = \rho V_S^2 \quad (2.17)$$

### 2.4.2.2 Constante de Lamé

También conocida como como primer parámetro de Lamé aunque no tiene una interpretación física directa o simple se usa en conjunto con el módulo de cizalla para caracterizar por completo el comportamiento elástico de un material isotrópico a través de la ley de Hooke, una ecuación constitutiva para un material homogéneo e isotrópico. Dentro de las ecuaciones disponibles para la estimación de este parámetro, se usó la que se presenta a continuación:

$$\lambda = \rho(V_p^2 - 2V_s^2) \quad (2.18)$$

### 2.4.2.3 Relación de Poisson

Es conocida como la tasa de la tensión trasversal o lateral a la tensión axial representada como:

$$\nu = \frac{\varepsilon_{lat}}{\varepsilon_{ax}} \quad (2.19)$$

También puede expresarse en función de las velocidades de onda S y de onda P usando la siguiente ecuación:

$$\nu = \frac{1}{2} \left( \frac{V_p^2 - 2V_s^2}{V_p^2 - V_s^2} \right) \quad (2.20)$$

### 2.4.2.4 Módulo de Young

Define el comportamiento de un material elástico según la dirección en el que se le aplique una fuerza, representada por la siguiente relación esfuerzo-deformación:

$$E = \frac{\sigma}{\varepsilon} \quad (2.21)$$

Al igual que los parámetros elásticos descritos anteriormente, es posible representar el módulo de Young en función de registros de pozos:

$$E = \frac{\rho V_S^2 (3V_P^2 - 4V_S^2)}{V_P^2 - V_S^2} \quad (2.22)$$

## 2.5 Modelo convolucional de la traza sísmica

El modelo matemático más sencillo para describir el proceso sísmico de reflexión de ondas sísmicas es el que está basado en el proceso de convolución (Regueiro, 2007): el concepto que una traza sísmica  $f(t)$  puede ser representado por la convolución de una ondícula  $w(t)$  con una función de reflectividad  $r(t)$  más un ruido aleatorio  $n(t)$ :

$$f(t) = w(t) * r(t) + n(t) \quad (2.23)$$

El modelo es una consecuencia del concepto de que cada onda reflejada causa su propio efecto en cada geófono (o hidrófono) independiente de otras ondas que afectan los mismos geófonos y que la respuesta del geófono es simplemente la suma (superposición lineal) de los efectos de todas las ondas.

La función  $f(t)$  puede discretearse y definirse como:

$$f_i(t) = \sum_{j=1}^N r_j W_{i-j+1} + n_i \quad (2.24)$$

Donde  $f_i$  representa la amplitud de la señal de la muestra  $i$  para una ondícula  $W_{i-j+1}$  y una reflectividad  $r_j$  a cero offset en el tope de la capa  $j$ , dado un ruido aleatorio  $n_i$ , siendo  $N$  el número

de capas involucradas (Lacedre Ricciardi, 2014). Este proceso puede expresarse esquemáticamente en la Figura 2.6.

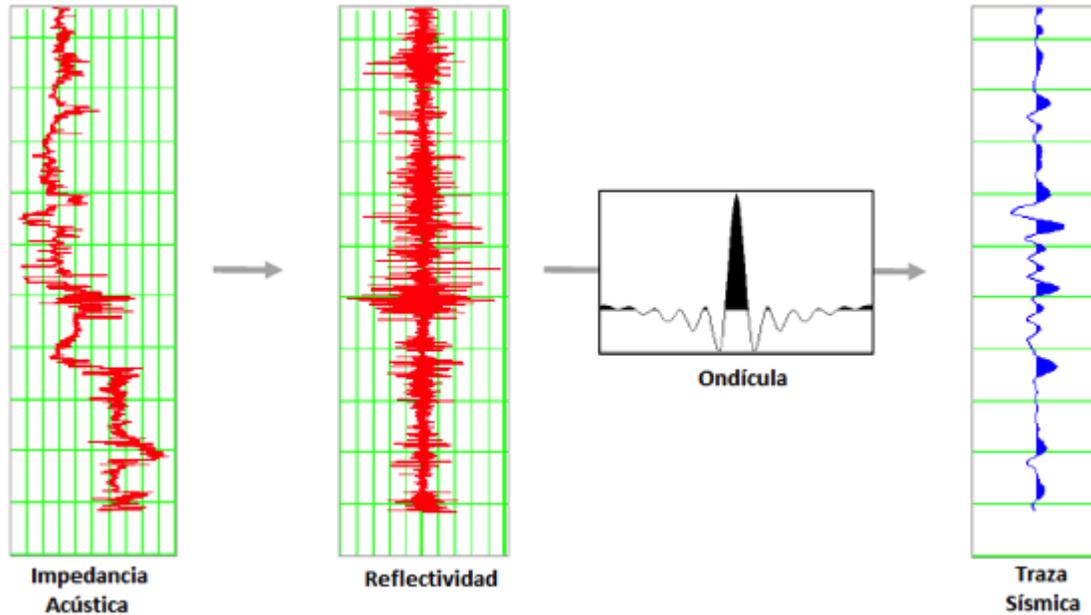


Figura 2.6: Esquema del modelo convolucional de la traza sísmica (Tomado de (Lacedre Ricciardi, 2014)).

Debido a que la serie de reflectividades está compuesta por funciones deltas, se puede interpretar como la superposición de las ondículas ubicadas en cada uno de los reflectores, escaladas por los valores de los coeficientes de reflexión correspondientes (Regueiro, 2007), expresada por la siguiente ecuación:

$$r_j = \frac{Z_{i+1} - Z_i}{Z_{i+1} + Z_i} = \frac{\rho_{i+1} * V_{p_{i+1}} - \rho_i * V_{p_i}}{\rho_{i+1} * V_{p_{i+1}} + \rho_i * V_{p_i}} \quad (2.25)$$

Donde:

Z: impedancia acústica

$\rho_i$ : el valor de la densidad del estrato suprayacente.

$V_{p_i}$ : el valor de la velocidad de onda P del estrato suprayacente.

$\rho_{i+1}$ : el valor de la densidad del estrato infrayacente.

$V_{p_{i+1}}$ : el valor de la velocidad de onda P del estrato infrayacente.

## 2.6 Resolución vertical sísmica

La resolución vertical de un registro sísmico puede definirse como la menor distancia vertical (espesor) entre dos interfaces litológicas que produce dos reflexiones discretas, las cuales pueden separarse visualmente (Regueiro, 2007). Este espesor, puede expresarse teóricamente de acuerdo a la expresión (2.26), relacionada con dos parámetros fundamentales del modelo convolucional: *la velocidad de propagación* en el material (componente geológico) y con el *contenido de frecuencia* o *ancho de banda* de la ondícula (componente sísmico) (Regueiro, 2007).

$$\lambda = \frac{V}{f} \quad (2.26)$$

Este valor de espesor, tiene un “límite de resolución” que corresponde teóricamente, y de forma aproximada, a un cuarto de la longitud de onda ( $\lambda/4$ ), espesor que también se conoce como el “punto de entonación (sintonía) sísmica”.

## 2.7 Atributos sísmicos

Las atributos sísmicos son cualquier medición de la data sísmica que nos ayuda a mejorar visualmente o cuantificar las características de interés interpretación (Chopra & Marfurt). Esta medición está basada usualmente en mediciones de tiempo, amplitud, frecuencia y/o atenuación. Generalmente, las mediciones basadas en tiempo se relacionan a rasgos estructurales, aquellas

basadas en la amplitud y frecuencia se corresponden a rasgos estructurales que son usados para la caracterización de yacimientos (Sheriff, 2002).

Los atributos son útiles para extender la data sísmica con alguna propiedad física de interés, siendo de gran utilidad en la mayoría de los casos para ver rasgos, relaciones y patrones que de otra manera no podrían ser vistos (Sheriff, 2002).

## **CAPÍTULO 3**

### **MARCO METODOLÓGICO**

La data sísmica y los registros de pozos fueron donados a la Universidad Simón Bolívar, los cuales fueron adquiridos en el Campo Blackfoot, distrito Alberta ubicado en Canadá en el año 1996 por el proyecto CREWES de la Universidad de Calgary. Para el análisis de estos datos, fue usado Petrel, un software con herramientas geológicas y geofísicas, y el lenguaje de programación de código abierto Python, debido a su rapidez y practicidad. Puesto que se contaba con muchos pozos y una variedad de atributos sísmicos, se les tenía que hacer un trabajo arduo y se trató de optimizar y automatizar cada una de la operaciones necesarias para generar los resultados, tratando de crear la cantidad de código necesario para ser reusado en la mayoría de los procedimientos involucrados.

La principal herramienta usada para la realización de los códigos fue IPython, una librería de Python que permite generar cuadernos interactivos que soportan código, texto, expresiones matemáticas, incrustación de gráficos y otros tipos de recursos (Pérez & Granger, 2007). Además, Petrel permitió realizar los amarres de pozos, la interpretación de los horizontes de interés, la generación de atributos sísmicos y la extracción de la información referentes a esos atributos que seguían la trayectoria de los pozos. En la Figura 3.1, se muestra de manera esquemática, las etapas principales involucradas en este trabajo de investigación, que tuvieron como base el flujo de trabajo estadístico para el análisis petrofísico definido por (Avseth, Mukerji, & Mavko, 2010).

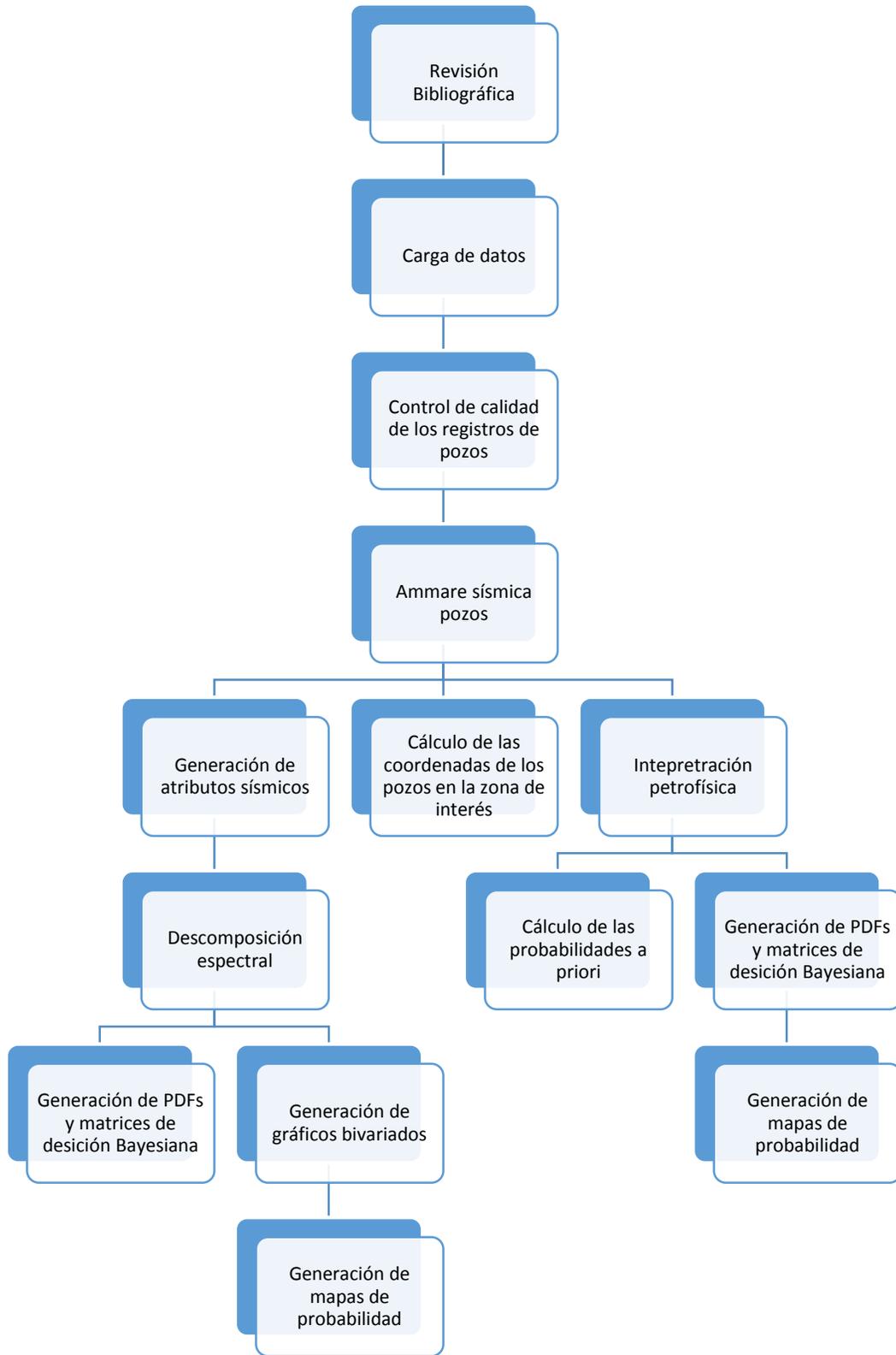


Figura 3.1: Representación esquemática de las etapas principales involucradas en este trabajo de investigación

### 3.1 Carga de datos sísmicos

La sísmica proporcionada (con el nombre de VERTMIG) tenía errores e información faltante en su cabecera, por lo tanto, para cargarla satisfactoriamente se debió establecer los valores asociados a los inlines y crosslines respectivos al patch de interés, como bien se sabe, el patch Glauconítico. Es por ello que en la Figura 3.2 se muestra la posición los inlines y crosslines que ayudaron a una correcta selección. El rectángulo con bordes oscuros (a la izquierda) corresponde al área de interés y la zona de la derecha sin resaltar corresponde al Patch Beaverhill Lake.

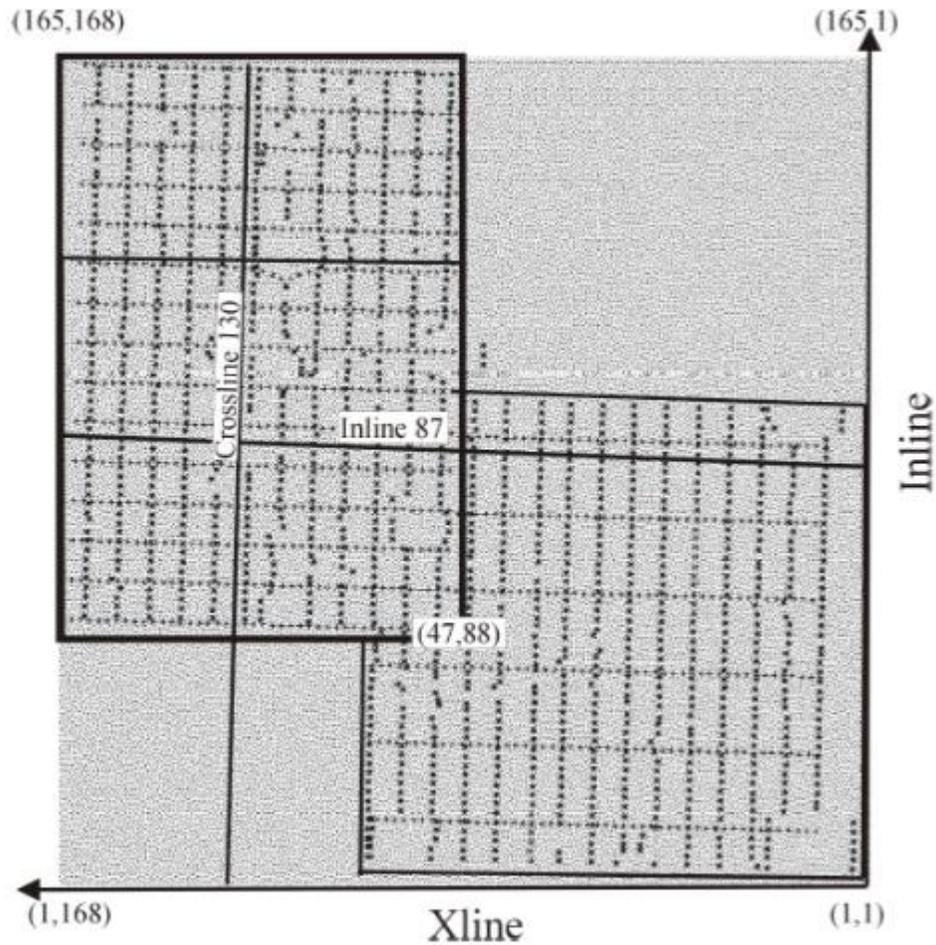


Figura 3.2: Posición relativa de inlines y crosslines del patch Glauconítico en el Campo Blackfoot (rectángulo negro y resaltado de la izquierda). Tomado de (Lu & Margrave, 1998)

Para el Patch Glauconítico el rango de inlines va de 47 a 165 y el de los crosslines de 88 a 168, estando ubicada esta información en los bytes 193 y 197 respectivamente. Además, la carga de información debe iniciarse en el byte 105 para no tener problemas de desfase de la sísmica en tiempo (Lacedre Ricciardi, 2014). El byte 73 tiene la información de las coordenadas en X con un rango de 348774m a 346409m y el byte 77 tiene la información asociada a las coordenadas en Y yendo desde 5644510m hasta 5648143m. En la Tabla 3.1 se muestran las coordenadas UTM de los vértices del Patch Glauconítico necesarios para ubicar espacialmente el volumen sísmico.

	Inline	Crossline	X(m)	Y(m)
NE	47	168	348868	5648078
NW	165	168	346409	5648143
SE	47	88	348774	5644510
SW	165	88	346316	5644575

Tabla 3.1: Coordenadas UTM, zona 12U, de los vértices del Patch Glauconítico.

### 3.2 Carga de los registros de pozos

En el Campo Blackfoot, específicamente en la zona de estudio, el *Patch* Glauconítico, existen información petrofísica de 12 pozos, los cuales llevan los nombres: 01-08, 01-17, 04-16, 05-16, 08-08, 09-05, 09-17, 11-08, 12-16, 13-16, 14-09 t 16-08. En la Tabla 3.2 se encuentran las coordenadas UTM en la zona 12U asociadas a estos pozos y en la Figura 3.3 su representación espacial de las coordenadas de sus bases.

Pozo	TOPE - UTM(m)		BASE - UTM(m)	
	X	Y	X	Y
01-08	347650.00	5644633.40	347512.1	5644916.7
01-17	347708.20	5646195.60	347543.9	5646522.7
04-16	347678.20	5646196.50	347922.1	5646449.2
05-16	347733.30	5647573.90	347852.6	5646909.8
08-08	347398.40	5645690.30	347525.9	5645264.6
09-05	347674.00	5644632.70	347496.4	5643927.9
09-17	347733.70	5647587.90	347517.4	5647388.1
11-08	347384.30	5645690.80	346930.4	5645671.7
12-16	347733.50	5647580.90	347944.0	5647387.8
13-16	348368.20	5648288.40	348222.7	5647809.2
14-09	348032.40	5646152.00	348231.6	5646169.8
16-08	347720.20	5646195.20	347565.7	5646066.9

Tabla 3.2: Coordenadas UTM, zona 12U, de los pozos proporcionados

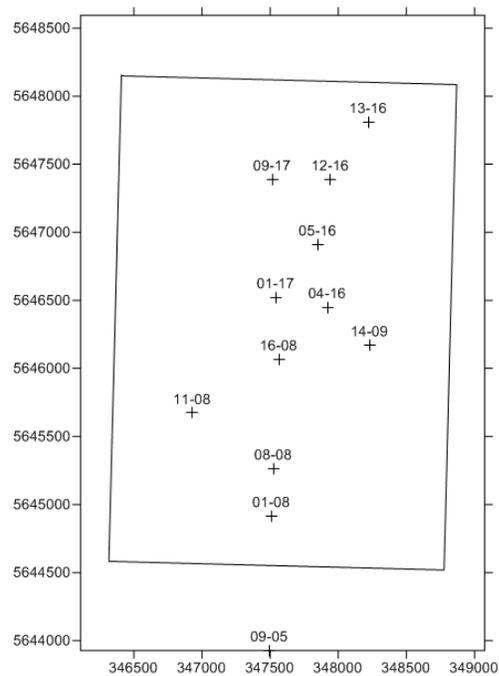


Figura 3.3: Ubicación de las bases de los pozos en el Campo Blackfoot.

Dentro de los registros disponibles (no en todos los pozos) se encuentran: Porosidad por densidad, Sónico P, Sónico S, Gamma Ray, Resistividad profunda, Resistividad media, Porosidad sónica, Densidad Bulk, Micro esféricamente enfocado y Potencial Espontáneo. La Tabla 3.3 contiene un inventario de los registros disponibles para cada pozo, indicando las celdas de color verde la disponibilidad del registro y las celdas de color rojo, el caso contrario. Debido a la poca variedad de registros, los pozos 04-16 y 12-16 fueron descartados para su interpretación petrofísica.

Nombre	Profundidad	Porosidad por densidad	Sónico P	Sónico S	Gamma Ray	Resistividad Profundidad	Resistividad Media	Porosidad sónica	Densidad Bulk	Micro Esféricamente enfocado	Potencial Espontáneo
	DEPTH	DPSS	DT	DTS	GR	ILD	ILM	NPSS	RHOB	SFL	SP
01-08	X	X	X	n/a	X	X	X	X	X	X	X
01-17	X	X	X	n/a	X	X	X	X	X	X	X
04-16	X	n/a	X	X	n/a	n/a	n/a	n/a	X	n/a	n/a
05-16	X	X	X	n/a	X	X	X	X	X	X	X
08-08	X	X	X	X	X	X	X	X	X	X	X
09-05	X	X	X	n/a	X	X	X	X	X	X	X
09-17	X	X	X	X	X	X	X	X	X	X	X
11-08	X	X	X	n/a	X	X	X	X	X	X	X
12-16	X	n/a	X	X	X	n/a	n/a	n/a	X	n/a	n/a
13-16	X	X	X	n/a	X	X	X	X	X	X	X
14-09	X	X	X	n/a	X	X	X	X	X	X	X
16-08	X	X	X	n/a	X	X	X	X	X	X	X

Tabla 3.3: Inventario de registros disponibles en los pozos del Campo Blackfoot.

En los reportes y tesis del proyecto CREWES, se hace una referencia importante a tres pozos: 08-08, 09-17 y 11-08. El pozo 08-08 es un productor localizado en una zona con anomalías altas de velocidad de Onda S (Todorov, 2000) y con presencia de gas desde el “Canal Superior” y petróleo desde el intervalo “Canal inferior Glauconítico” (Larsen, 1999). El pozo 09-17 es un pozo no productor (Larsen, 1999) localizado en una área de relativa baja impedancia con anomalías bajas de velocidad de onda S (Todorov, 2000). Y el pozo 11-08 contiene datos de velocidad de onda P desde la superficie al tope del Canal Glauconítico.

### 3.3 Cálculos de coordenadas a una cierta profundidad en pozos desviados

Lo pozos proporcionados presentaban una desviación en promedio de 14 grados, la cual generaba un cambio de trayectoria que provocaba una alteración de las coordenadas en la información asociada a los registros petrofísicos en cada intervalo de profundidad. Debido a esto, las coordenadas de nuestra zona de estudio son distintas a las reportadas por los toques y las bases de los registros petrofísicos (Figura 3.4). Por lo tanto, fue necesario recurrir a métodos matemáticos para hallar las coordenadas de la información petrofísica a la profundidad de nuestra zona de interés.

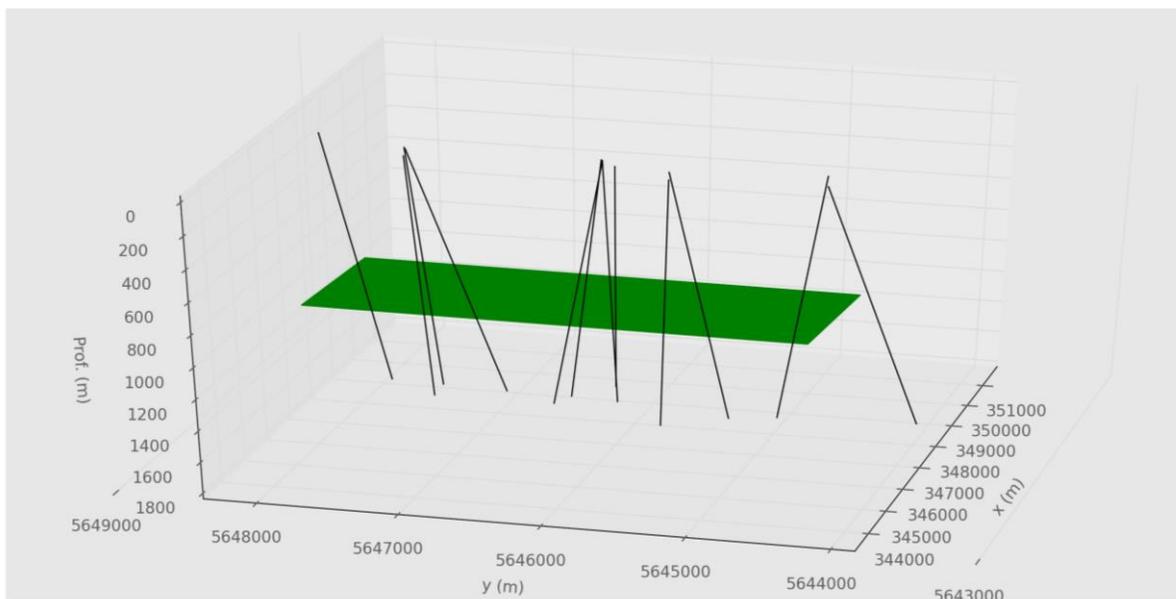


Figura 3.4: Representación espacial de los pozos desviados, intersectados por la profundidad media de la zona de interés (rectángulo verde)

Supongamos un pozo desviado, teniendo sobre él dos puntos cualesquiera  $P_o(P_{ox}, P_{oy}, P_{oz})$  y  $P_1(P_{1x}, P_{1y}, P_{1z})$ , y una zona de interés representada por un plano, el cual tiene como vector normal  $\vec{n} = (n_x, n_y, n_z)$  que puede o no estar normalizado y un punto sobre éste representado por  $P_{co}(P_{cox}, P_{coy}, P_{coz})$  (Figura 3.5).

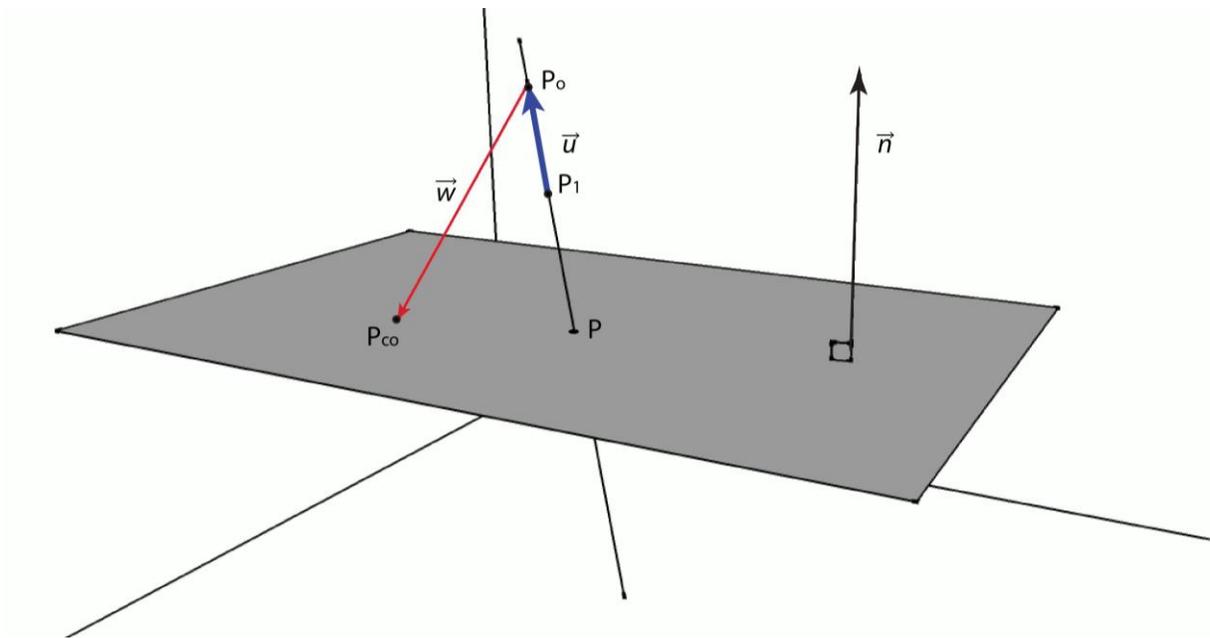


Figura 3.5: Representación de un pozo desviado  $\vec{u}$  intersectado por un plano.

Definimos el pozo con el vector director  $\vec{u}$ :

$$\vec{u} = (P_1 - P_o) \quad (3.1)$$

y un vector  $\vec{w}$  que conecte un extremo del pozo con uno de los extremos del vector director del pozo:

$$\vec{w} = (P_o - P_{co}) \quad (3.2)$$

Es necesario verificar si el pozo no es paralelo al plano, esto se realiza probando  $\vec{u} \cdot \vec{n} = 0$ , lo que significa que estos vectores son ortogonales. Si  $\vec{u}$  y  $\vec{n}$  no son paralelos, entonces se intersectan en un único punto. El valor de este punto se calcula usando la siguiente ecuación (Coxeter, 1989):

$$P = \vec{u}_2 + P_o \quad (3.3)$$

siendo  $\vec{u}_2$  calculado como:

$$\vec{u}_2 = \vec{u} \cdot s \quad (3.4)$$

donde:

$$s = \frac{-\vec{n} \cdot \vec{w}}{\vec{u} \cdot \vec{w}} \quad (3.5)$$

### 3.4 Interpretación petrofísica

Para caracterizar litológicamente nuestros registros de pozos, fue necesario realizar una interpretación petrofísica basada en el procedimiento descrito por (Mendoza & Landa, 2005) y considerando los fundamentos teóricos de (Glover, 2014).

La interpretación petrofísica fue enfocada entre los topes MANN (Blairmore - Mannville Superior) y GLCSS (Unidad de arena porosa del Canal Glauconítico), en las cuales se incluían las arenas productoras. Siguiendo la recomendación de (Larsen, 1999), una vez extraídos los registros, con el fin de remover las fuertes reverberaciones, fue aplicado un filtro mediano con una ventana de 11 muestras que representaban aproximadamente un muestreo de 1.2 metros.

Luego, el volumen de arcilla fue obtenido usando la ecuación (2.8), tomando valores diferentes de Gamma Ray sucio y limpio para cada pozo, para posteriormente, con un Vshale de corte igual a 0.3. Las zonas con valores menores fueron clasificados como permeables (arenas limpias) y zonas con valores mayores como impermeables (arenas sucias y posibles lutitas).

Para la identificación de las zonas con alto contenido de agua en las arenas limpias, fue necesario calcular la saturación de agua usando las ecuaciones (2.13) y (2.12); se asumió un valor de 0.7 de saturación de agua como valor de corte, aquellos que fuesen mayor a este corte eran clasificados como posible acuíferos.

La posibles zonas de gas fueron identificadas con los registros de porosidad neutrón ( $\phi_N$ ), porosidad por densidad ( $\phi_D$ ) y resistividad ( $R_T$ ), identificando los comportamientos asociados a una disminución de  $\phi_N$  y un aumento de  $\phi_D$  con una diferencia entre ellos mayor a 0.7 unidades con valores altos de  $R_T$  asociados.

Una vez identificados las zonas con fluidos, fue construido un nuevo vector de litología para cada pozo, el cual representaba la clasificación litológica por fluidos derivada de la interpretación petrofísica.

Por último, dependiendo de los registros disponibles, fueron calculadas las impedancias acústicas y relativas (Sección 2.5.1) y los parámetros elásticos (Sección 2.5.2) asociados a cada pozo.

### **3.5 Cálculos de probabilidades a priori**

Para determinar la probabilidad a priori asociada a una litología fue necesario primero calcular su espesor total que ésta representa en nuestra zona de interés expresada por la suma de aquellos espesores asociados a las ocurrencias de nuestra litología (Figura 3.6):

$$E_{Total} = E_1 + E_2 + \dots + E_N = \sum_{i=1}^N E_i \quad (3.6)$$

Luego, esta suma total fue dividida por el espesor total de nuestra zona  $E_Z$  para dar resultado a la fracción de ocurrencia que representa la probabilidad a priori de esta litología. Todo este procedimiento fue aplicado a cada litología identificada a cada pozo, ya que en ellos se presentaban una distribución litológica distinta.

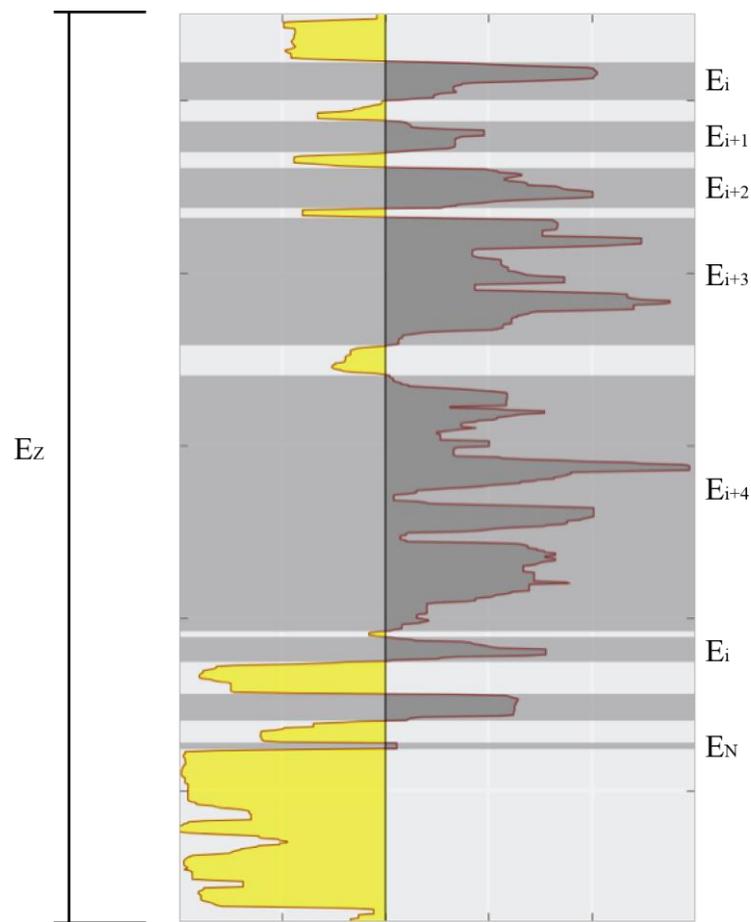


Figura 3.6: Espesores de un grupo litológico (zona sombreada) para el cálculo de su probabilidad a priori.

### 3.6 Generación de las Funciones de densidad de probabilidad

Para generar las funciones de densidad de probabilidad (PDFs) fue necesario tener clasificada nuestra propiedad a ser evaluada con dos grupos (litologías o fluidos) como mínimo. Luego de asegurar esta condición, se separaron estos grupos y se construyeron histogramas que mostraran la distribución de estos datos (Figura 3.7a) que nos permitieron construir las PDFs derivados de los histogramas con un ajuste gaussiano estimado (Figura 3.7b). Finalmente, para tener unas PDFs optimas ajustadas a condiciones litológicas iniciales, se multiplicaron los grupos con sus probabilidades a priori (Figura 3.7c).

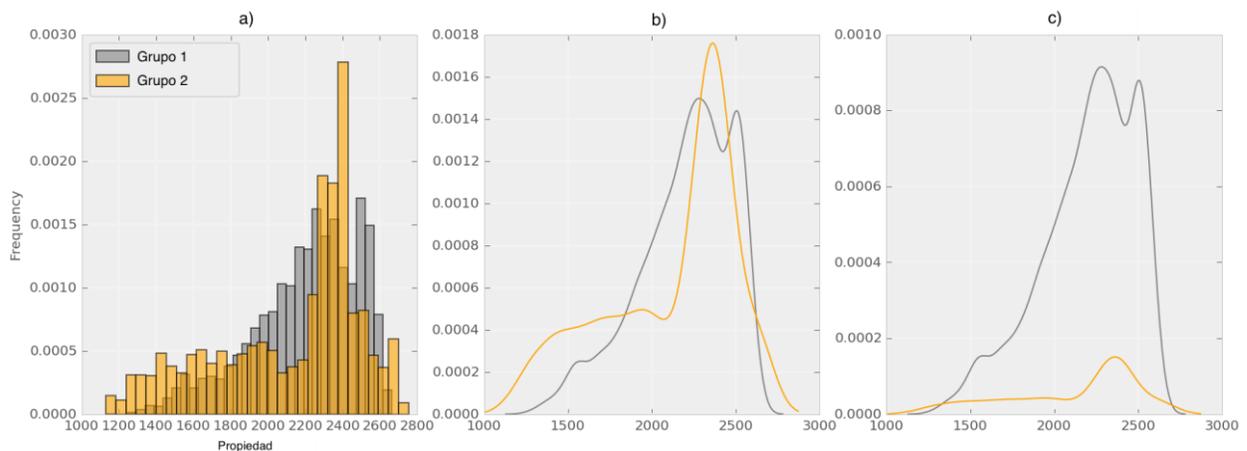


Figura 3.7: **a)** Histogramas asociados a dos grupos pertenecientes a un propiedad. **b)** Funciones de densidad de probabilidad que representan el comportamiento de los histogramas en a). **c)** Funciones de densidad de probabilidad de a) multiplicadas por la probabilidad a priori de cada grupo.

Un procedimiento similar fue aplicado para los gráficos bivariados con la salvedad de que se estaba trabajando con datos en dos dimensiones. Primero, con las propiedades seleccionadas se generó un histograma 2D para cada litología, luego tomando esas matrices generadas, se procedió a generar mapas de contornos que representaran la distribución estadísticas, como se puede observar en la Figura 3.8.

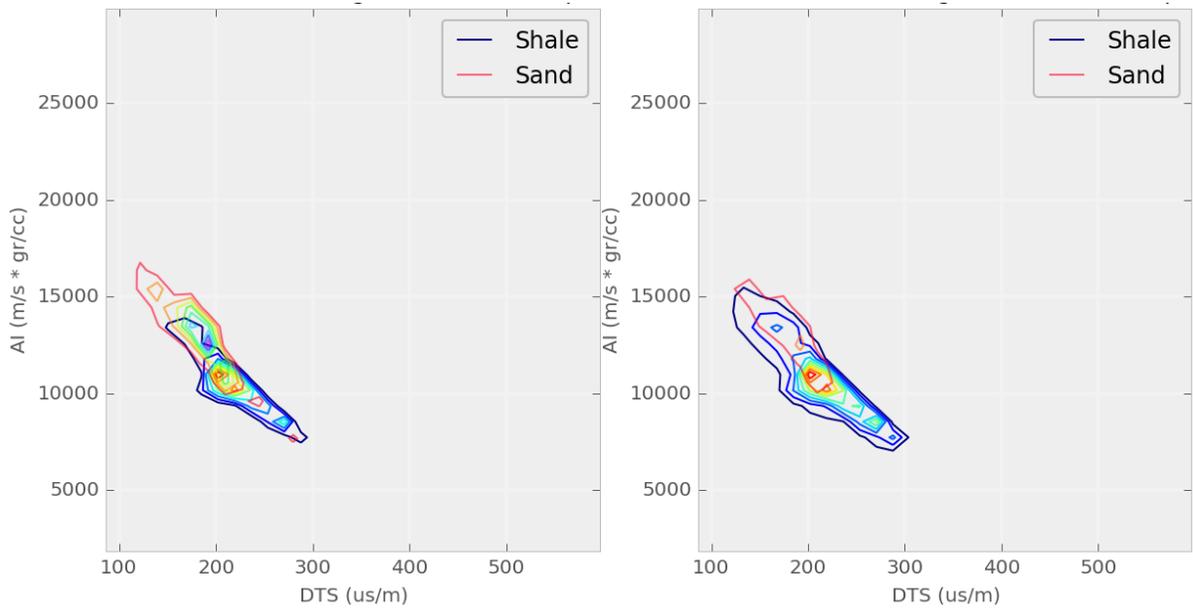


Figura 3.8: Gráficos bivariados que representan la distribución estadísticas de dos litologías para dos propiedades distintas.

### 3.7 Generación de la matriz de decisión Bayesiana

Una vez generadas las curvas de probabilidad, fue necesario conocer qué porcentaje de cierta propiedad fue clasificada como verdadera haciendo uso de las matrices de decisión bayesiana. Una de las formas encontradas más eficientes y automatizada fue la de generar un código que hiciera todo el trabajo siguiendo el siguiente procedimiento:

- Para dos curvas probabilísticas (Figura 3.9a) que puede representar distintas litologías, fue calculada el área correspondiente usando el método establecido por la suma de Riemann. Este valor obtenido corresponde a la máxima clasificación de la propiedad clasificada como tal.
- Tomando como referencia la curva 1, fue comparado cada valor de la misma con los valores de la curva 2, si estos valores eran menores se igualaban a cero, si ocurría lo contrario no se

hacía ninguna operación matemática (ver Figura 3.9b). Lo mismo se realizaba para la curva 2.

- Luego, se tomaban las curvas resultantes (curva 1.1 y curva 2.1) y se le calculaba el área bajo ellas. Este valor representa la fracción del área total que está siendo clasificada correctamente, es decir, para la curva 1, la fracción de la curva 1 que está siendo clasificada como curva 1.
- Posterior a esto, fue sencillo hallar la fracción areal de la curva que estaba siendo clasificada como la otra curva: restándole al área total la fracción areal que estaba siendo clasificada correctamente.
- Finalmente normalizando estos valores, se pudieron construir la matriz de decisión bayesiana correspondiente.

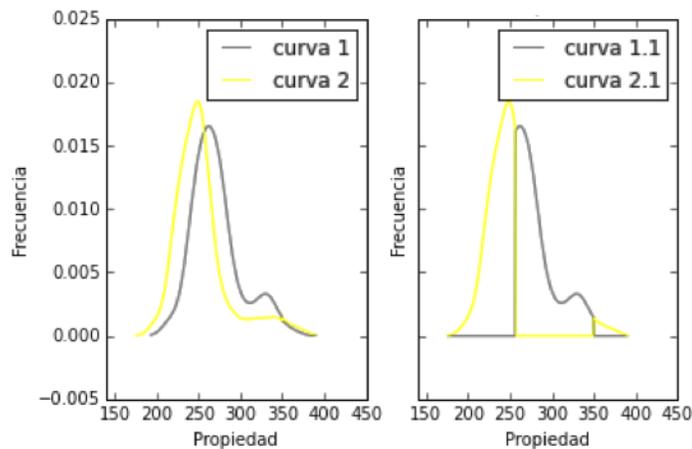


Figura 3.9: Curvas probabilísticas con para la generación de la matrices de decisión bayesiana. a) Curvas de probabilidad asociadas a una propiedad. b) Curvas de probabilidad cuyos valores son iguales a cero en aquellas zonas que estuviesen por debajo de la curva restante.

	Curva 1	Curva 2	
Curva 1	Fracción areal de la curva 1 clasificada como curva 1	Fracción areal de la curva 1 clasificada como curva 2	Área total de la curva 2
Curva 2	Fracción areal de la curva 2 clasificada como curva 1	Fracción areal de la curva 2 clasificada como curva 2	Área total de la curva 2

Tabla 3.4: Matriz de decisión bayesiana de ejemplo, resultante del procedimiento propuesto en este apartado.

### 3.8 Generación de mapas de probabilidad usando los atributos sísmicos

EL procedimiento para generar mapas de probabilidad asociados al porcentaje arena clasificado correctamente usando los atributos sísmicos fue similar al usado en el análisis petrofísico, pero un poco más extenso. El procedimiento utilizado fue el siguiente:

- Se identificaron los horizontes de interés en la sísmica usando el programa Petrel.
- Luego se generaron todos los atributos volumétricos posibles.
- Para cada pozo se hizo una clasificación bayesiana de la información de cada atributo en donde pasaba el pozo entre el tope y la base de los horizontes de interés, siguiendo el mismo procedimiento realizado en la clasificación estadísticas con las propiedades de los registros de pozos.
- Se extrajeron dos trazas correspondientes a dos atributos en una misma posición geográfica (ver Figura 3.10).
- Posteriormente, los valores extraídos de cada atributo a una misma profundidad (ver Figura 3.10) se llevaron a un gráfico bivariado correspondiente para determinar la litología dominante. Este grafico bivariado representa un promedio de todos los gráficos bivariados

asociados a cada pozo, tomando en cuenta la probabilidad a priori de cada litología. Todo esto se hace para cada intervalo de profundidad.

- Una vez obtenido la litología dominante a cada profundidad, se calculó el porcentaje de arena que ha sido clasificado como tal en esa posición a través en toda la columna de datos evaluada. Este porcentaje formará parte de un mapa del contenido de arena en nuestra zona de estudio, ya que es el resultado de un procedimiento que toma en cuenta la posición de la traza seleccionada.
- Finalmente, todo lo anterior descrito se hace para cada par de traza asociado a cada par de atributos posible.

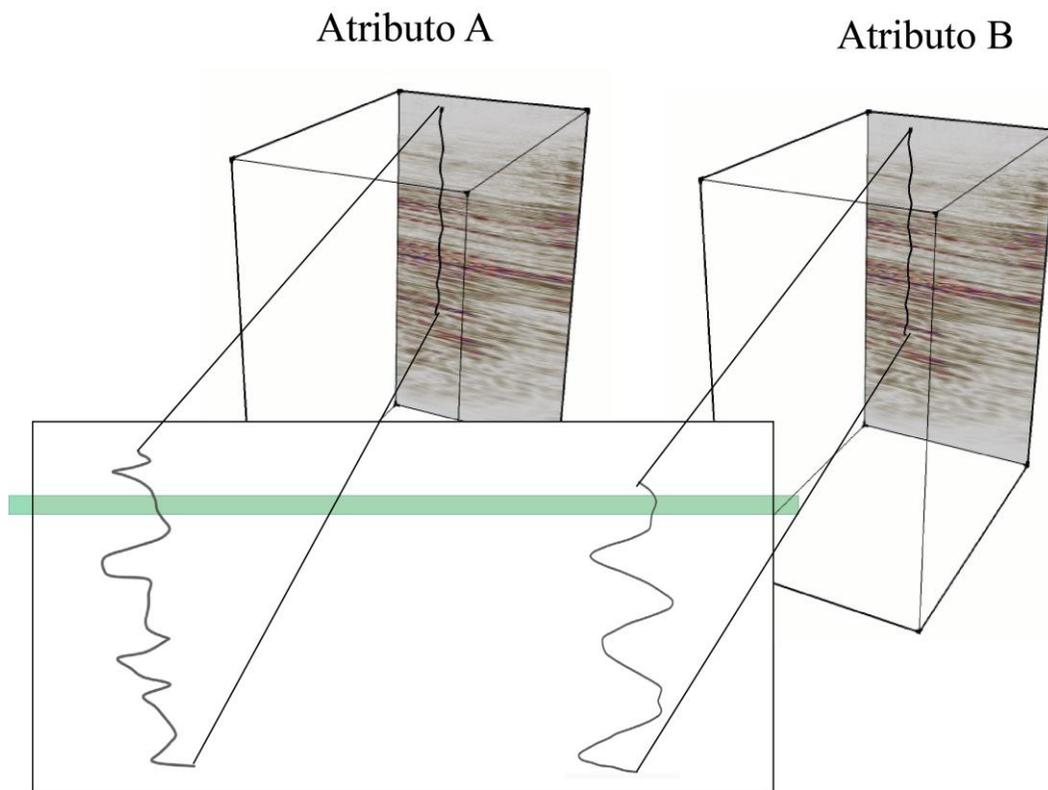


Figura 3.10: Representación de la extracción de valores de amplitud a cierta profundidad correspondientes a dos trazas de dos atributos.

## **CAPÍTULO 4**

### **RESULTADOS Y ANÁLISIS**

En las siguientes secciones se mostraran los resultados de cada fase descrita en el Capítulo 3, tomando como datos de entrada el volumen sísmico y los registros petrofísicos de cada pozo.

#### **4.1 Coordenadas de los pozos en la zona de estudio**

Los pozos presentaban en promedio una desviación de aproximadamente  $14^\circ$ , resultados que para nuestro espesor de estudio las coordenadas de estos pozos con respecto a las coordenadas en superficie reflejan una diferencia entre 30m y 700m. Esto resultó que el pozo 09-05 no entrara en la zona del Patch Glauconítico, siendo excluido de los análisis posteriores asociados a la información sísmica (Figura 4.1).

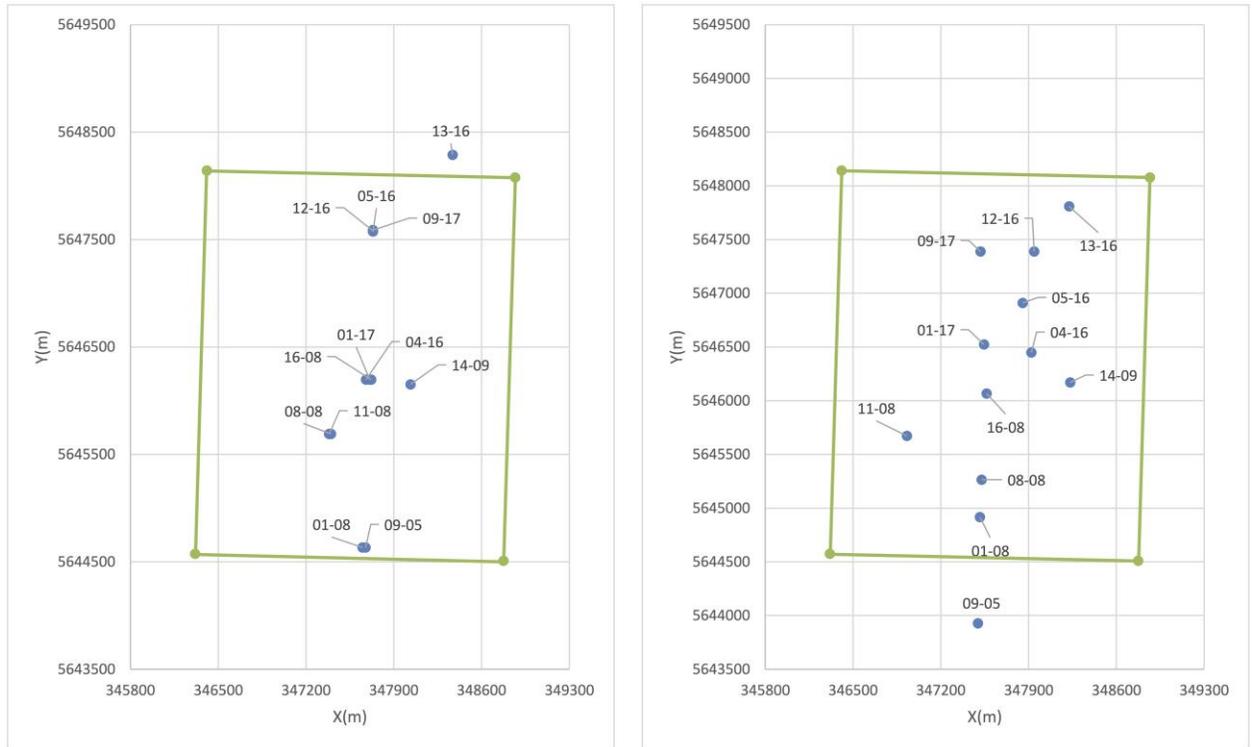


Figura 4.1: A la izquierda la representación de las coordenadas de los topos de cada pozo y a la derecha las coordenadas de los pozos en la zona de estudio.

## 4.2 Control de calidad de los registros de pozos

Durante la realización de la revisión de los datos disponibles se pudo constatar a través de pruebas con amarres de pozos y la revisión bibliográfica de los trabajos realizados por (Lacedre Ricciardi, 2014), (Massaad, 2014) y (Bautista, 2011) que el Kelly Bushing presentado en los archivos con información de los registros de pozos estaba incorrecto, estos valores deberían ser nulos.

Tal como se expresó anteriormente, el pozo 09-05 no atraviesa la sísmica proporcionada, por lo tanto no fue requerido para los análisis relacionados a la misma. Sin embargo, este pozo fue interpretado petrofísicamente. Otro detalle encontrado, fue que el pozo 04-16 no tenía disponible el registro de Rayos Gamma, siendo descartado de cualquier análisis ya que para la clasificación litológica realizada en este trabajo, ese registro era de vital importancia.

Por otro lado, los pozos 09-17 y 12-16 contaban con los registros sínicos P y sínico S, lo cual permitió la generación de los parámetros elásticos expresados en el Marco Teórico (página 22). A pesar de que el pozo 08-08 tenía disponible el registro de Sínico S, este no estaba disponible en nuestro espesor de estudio. Algo muy importante a resaltar, que independiente de los resultados obtenidos con los parámetros elástico (los cuales se detallan más adelante), estos resultados no son de gran utilidad ya que sólo pudieron obtenerse a través de dos pozos que no representaban geográfica ni geológicamente la distribución areal del resto de los pozos.

Fue necesario aplicarles a los registros de pozos a ser utilizaos un filtro mediano, como fue expresado en la Metodología (página 37). Todos los registros de pozos presentaban reverberaciones pero los pozos 09-05 y 16-08 representaban aquellas más pronunciadas. La Figura 4.2 presenta el registro de Rayos Gamma del pozo 16-08, obsérvese detalladamente como ciertos picos y las reverberaciones en todo el registro desaparecen al aplicarle el filtro ya mencionados, conservándose las tendencias y haciendo un poco más suavizadas las curvas con el fin mejorar la interpretación del registro en cuestión. Los parámetros de entrada del filtro fueron escogidos cuidadosamente, se realizaron varias pruebas y la que mejor generó resultados fue aquella que presentaba una ventana de escogencia para el filtro de 11 muestras, la cual representaba aproximadamente un muestreo de 1.2 metros.

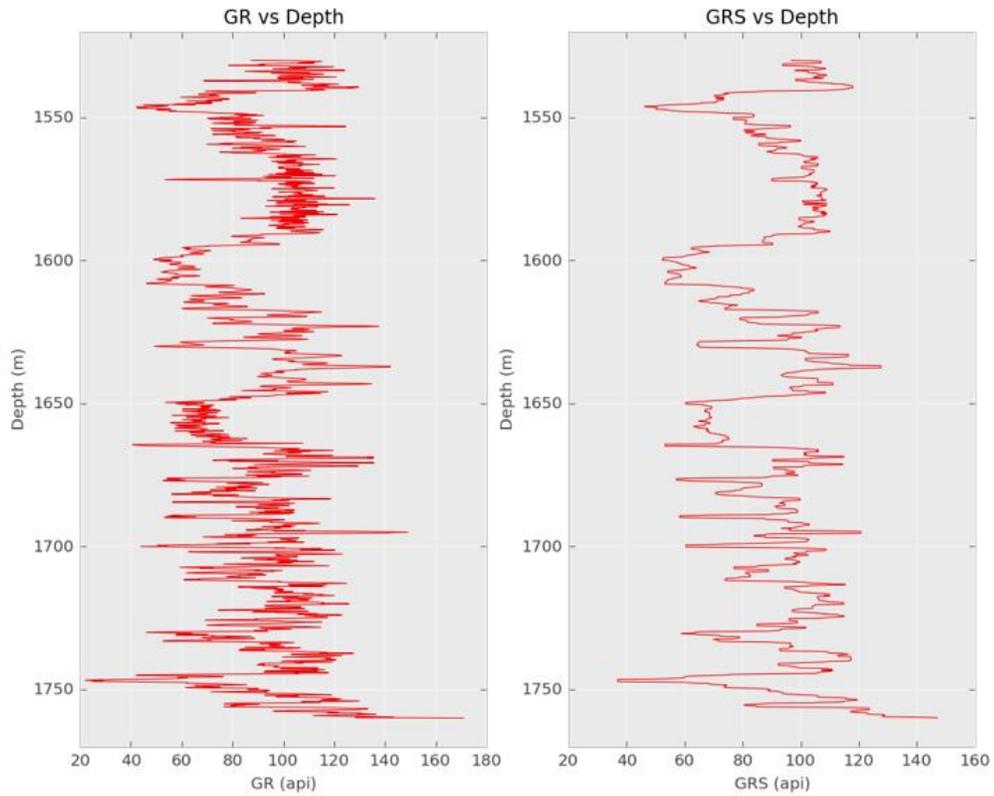


Figura 4.2: Registro de Rayos Gamma perteneciente al pozo 16-08. EL registro de la izquierda representa la data original sin ninguna modificación, mientras que el de la derecha fue mejorado con un filtro mediano con una ventana de 11muestras.

No solo se pudo mejorar los registros originales del pozo, sino que aquellos pseudo-registros generados para la interpretación petrofísica mostraban la información de una manera más clara, como puede observarse en la figura Figura 4.3. Para este caso, el registro de la derecha, evidencia una mejor delimitación de los espesores de arenas, permitiendo una identificación más acertada de los eventos que pueden estar presentes.

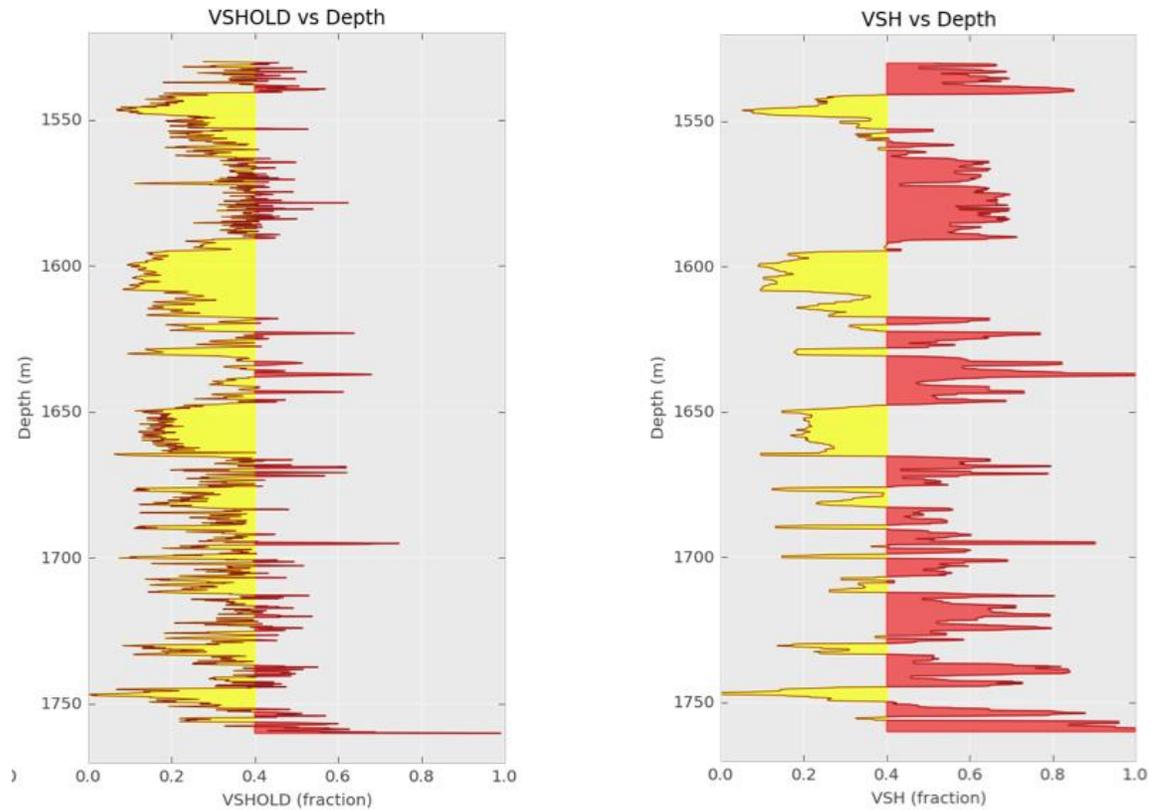


Figura 4.3: Registro de Volumen de arcilla perteneciente al pozo 09-05. El registro de la izquierda representa la data original sin ninguna modificación, mientras que el de la derecha fue mejorado con un filtro mediano con una ventana de 11 muestras.

### 4.3 Identificación de cuerpos arenosos y lutíticos

En la mayoría de los pozos, específicamente el 01-17, 05-16, 09-05, 09-17, 11-08, 12-16, 13-16, 14-09, presentaron intercalaciones de arenas y lutitas, un comportamiento esperado ya expuesto por (Hayes, et al., 1994). La Figura 4.4 representa dos registros de ejemplo de volúmenes de arcillas, se observa una sucesión de intercalaciones de arenas y lutitas relativamente uniformes, presentando en forma general espesores de arenas muy bajos en comparación a los espesores de las lutitas, lo que puede asociarse a un sistema de baja energía.

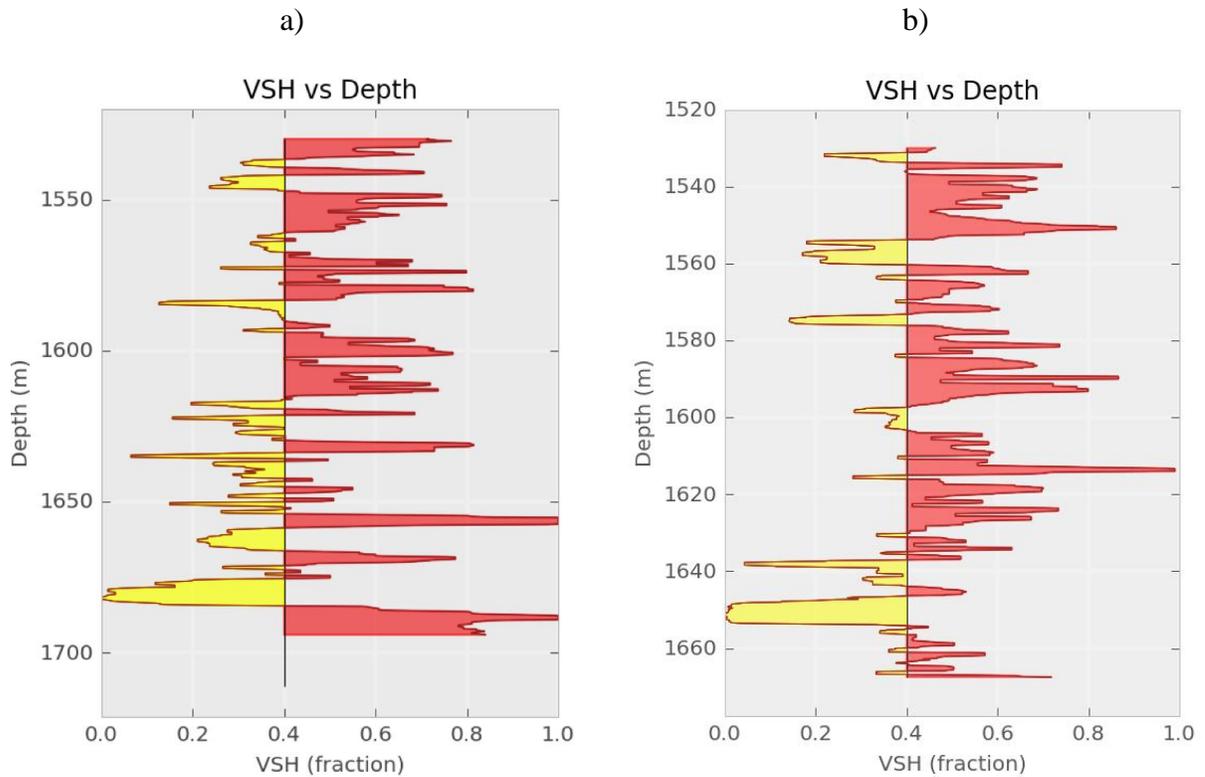


Figura 4.4: Registros de volúmenes de arcilla. a) Registros pertenecientes al pozo 09-17 b) Registros perteneciente al pozo 01-17.

Los pozos 01-08, 08-08 y 16-08 presentaban una intercalación de áreas y lutitas no pronunciada, y se destacaban por presentar un espesor de arena considerable entre 20m y 50m aproximadamente, cercano a los 1620m (Figura 4.5a), que podrían ser producto de los valles incisos presentes en la zona de estudio (Yang, et al., 1996). Además, el pozo 13-16 (Figura 4.5b), a pesar de presentar intercalaciones de arenas y lutitas, presentaba un cuerpo de arena con un espesor de aproximadamente 25m cercanos a los 1600m de profundidad. Algo similar pudo observarse en el pozo 05-16 (Figura 4.5c), un cuerpo arenosos presente 1730m y 1780m de profundidad, con contenido lutito, la cual podría ser una zona limo-arenosas interpretadas por (Yang, et al., 1996).

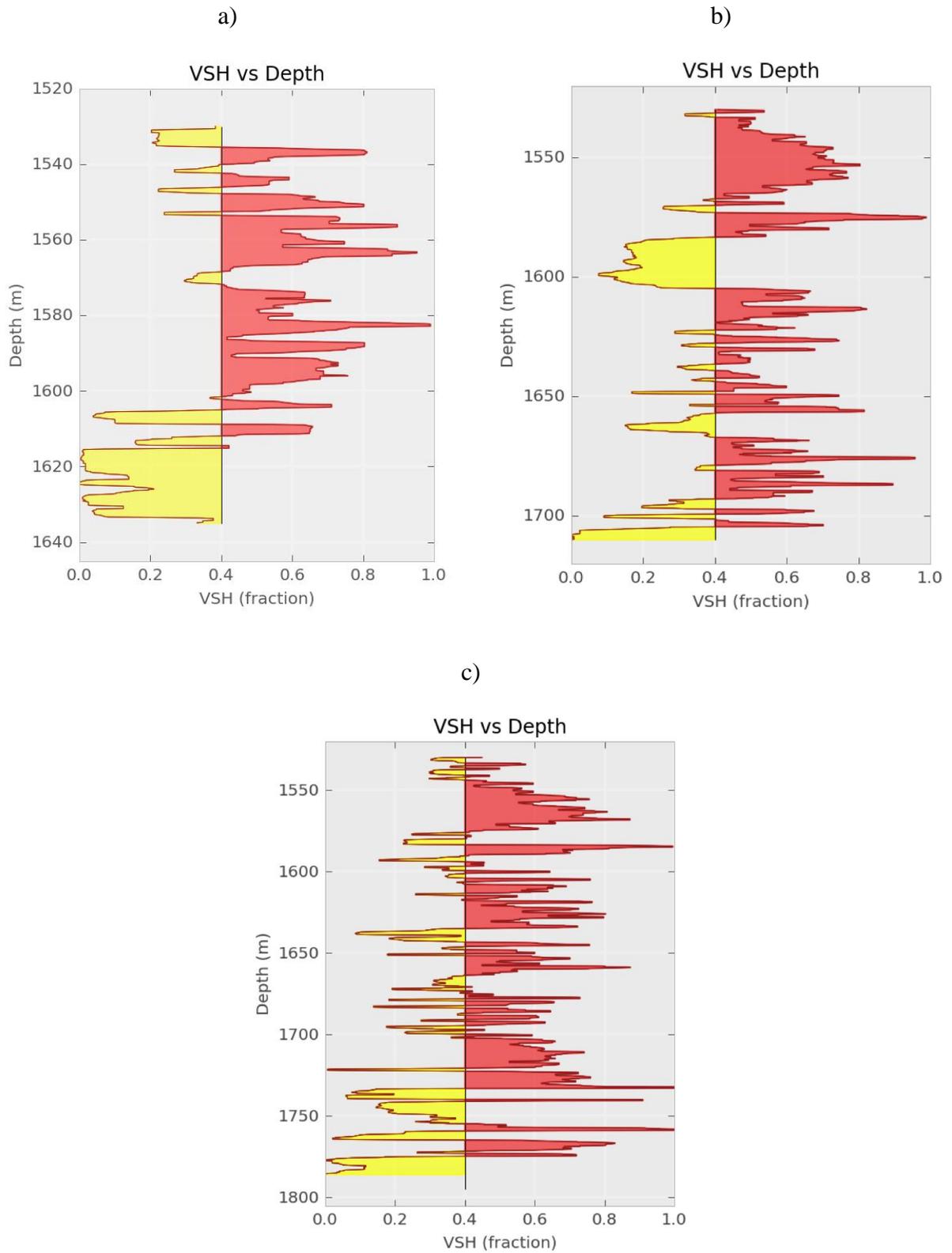


Figura 4.5: a) Registros de volumen de arcilla del pozos 01-08. b) Registro de volumen de arcilla del pozo 13-16. c) Registro de volumen de arcilla del pozo 05-16.

En términos generales, nuestra interpretación de los registros de rayos gamma se ajustó muy bien a aquellas realizadas por (Yang, et al., 1996). En la zona inferior de la Figura 4.6a se observa el grupo de pozo mencionados anteriormente que presentan un espesor de arena importante, los cuales coinciden con el canal de arena identificado en la Figura 4.6b. Además, se puede asumir también que lo identificado en el pozo 13-16 y 05-16 (Figura 4.6a) podrían deberse a cuerpos arenosos posiblemente aislados tal como como podemos evidenciar en la Figura 4.6b.

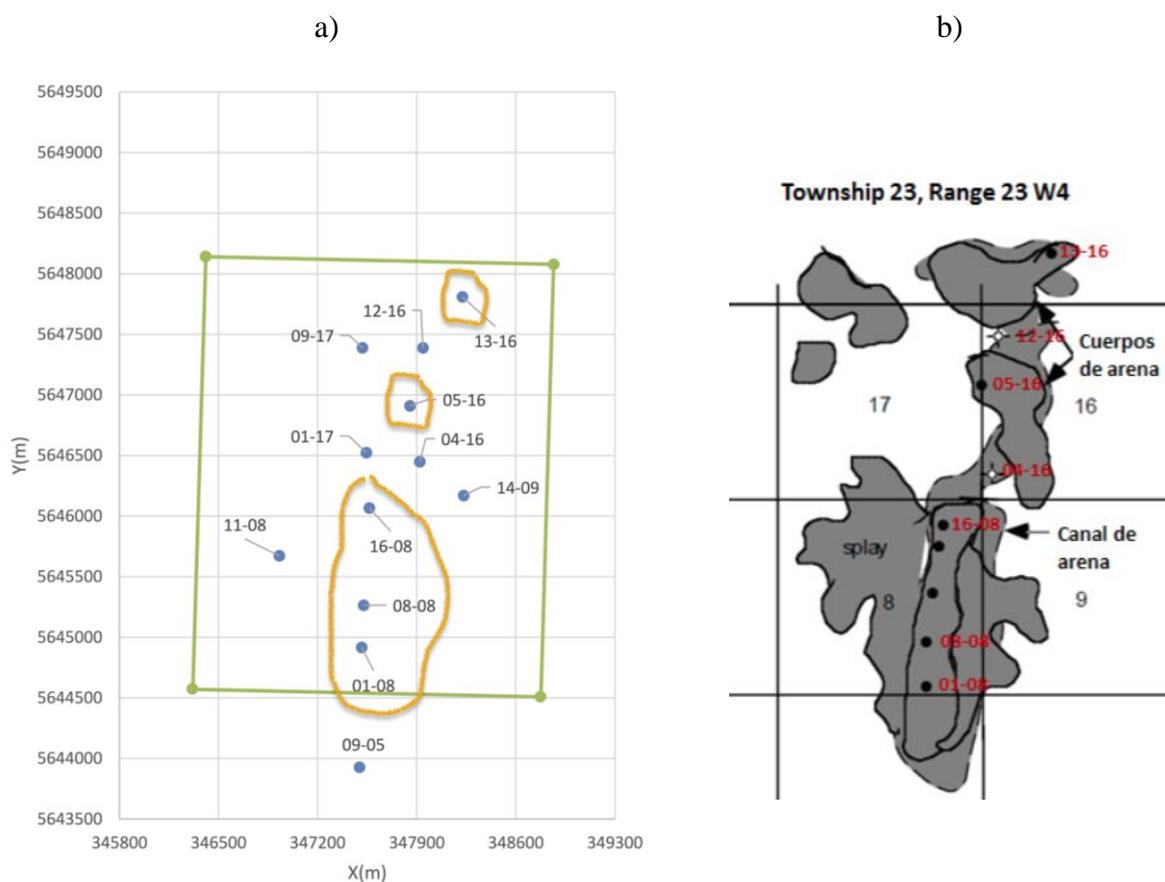


Figura 4.6: Interpretación de cuerpos de arena a nivel en la formación Glauconítica a) Cuerpos arenosos identificados con la interpretación de los registros de Rayos Gamma b) Interpretación realizada por (Yang, et al., 1996).

#### **4.4 Contenido de arena (Probabilidad a Priori)**

Para todos los pozos interpretados petrofísicamente, se les fue calculado el porcentaje de arena contenido en el espesor de interés, entre los topes MANN (Blairmore - Mannville Superior) y GLCSS (Unidad de arena porosa del Canal Glauconítico). Como se indicó previamente, los pozos 01-17, 05-16, 09-05, 09-17, 11-08, 12-16, 13-16, 14-09 presentaban una variedad de intercalaciones de arenas y lutitas, pero al estudiar cuantificar el contenido de arena se observó que la tendencia predominante es que estos pozos tienen una cantidad mayor de estratos arenosos (Figura 4.7). Observe bien que este mapa de contenido de arena puede o no representar la evidencia de algún cuerpo o procesos geológicos, por ejemplo, los pozos 13-16 y 05-16 que presentan cuerpos de arenas, tenían una cantidad mayor de estratos lutíticos, mientras que los pozos que representaban el posible canal de arena (Yang, et al., 1996), 16-08, 08-08 y 09-05, presentaron una cantidad de estratos lutíticos que iba desde el 50% y 5%. Por otro lado pozos 09-17, 12-16, 11-08 y 09-05 presentaron un contenido de arena entre el 70% y 80%.

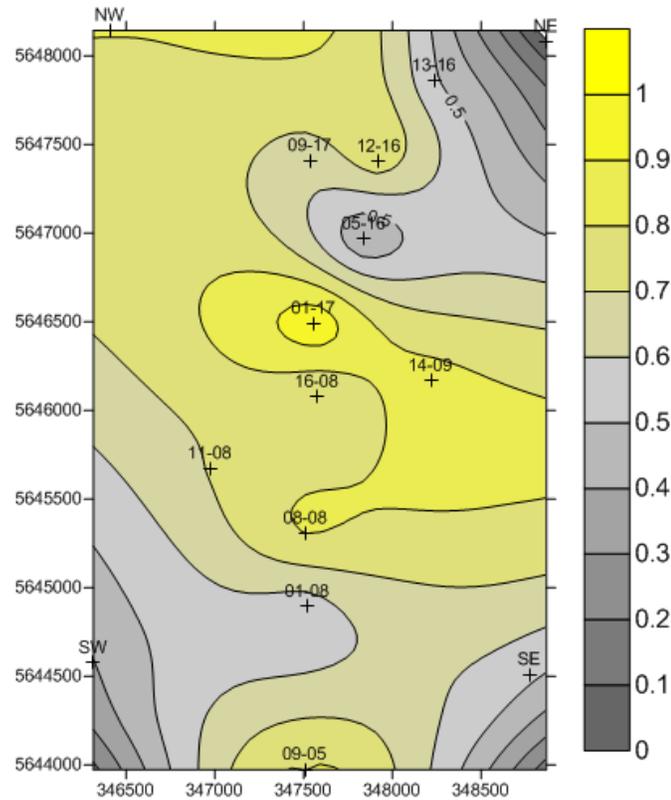


Figura 4.7: Probabilidad a priori (o contenido) de arenas, donde el valor máximo representa 100% de arena y 0% de lutita.

#### 4.5 Discriminación litológica

Se generaron curvas PDF para todos los registros y pseudo-registros en cada pozo que estuviesen asociados a diferencias litológicas. Para los pozos 01-08, 01-17, 05-16, 08-08, 09-05, 11-08, 13-16, 14-09 y 16-08 se generaron para el Sónico P, Densidad Volumétrica e Impedancia Acústica, teniendo estos tres registros una muy buena discriminación del contenido de arena y lutitas (como se verá más adelante). Los mismos registros se estudiaron para los pozos 09-17 y 12-16 teniendo también una buena clasificación, pero además, debido a que tenía disponible el registro Sónico S, se realizó una discriminación del mismo y de otros pseudo-registros: Impedancia Elástica, Módulo de Cizalla, Constante de Lamé, Relación de Poisson y Módulo de Young. Ninguno de estos registros generados a partir del Sónico S realizó una buena discriminación litológica, la tendencia predominante era que las curvas PDF de la arena y la lutita tenían el mismo comportamiento.

#### **4.5.1 Curvas PDFs Univariadas**

El comportamiento general observado fue que al tomar en cuenta el porcentaje de arena (la probabilidad a priori) asociado a cada pozo, los porcentajes de clasificación variaban significativamente.

##### **4.5.1.1 Sónico P**

El primer rasgo a identificar en las PDFs generadas (Figura 4.8, Figura 4.9 y Figura 4.10) es que al tomar en cuenta la probabilidad a priori de las arenas y lutitas, el alto de las curvas asociadas a las litología aumentaban y el alto de las curvas asociadas a arenas disminuía, ocasionando que en algunas zonas de estas PDF no se produjera una buena clasificación. Además se puede observar que las PDFs de cada pozo presentan un comportamiento aceptable para su uso, no tiene un comportamiento similar sobre un mismo punto.

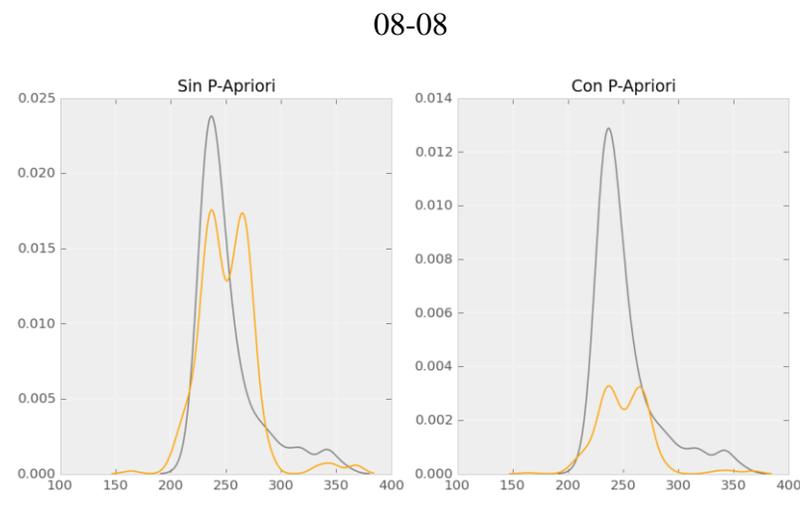
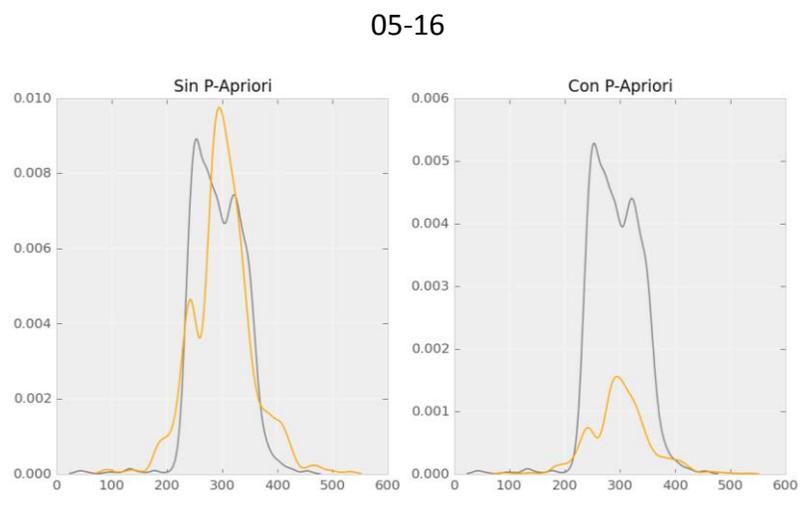
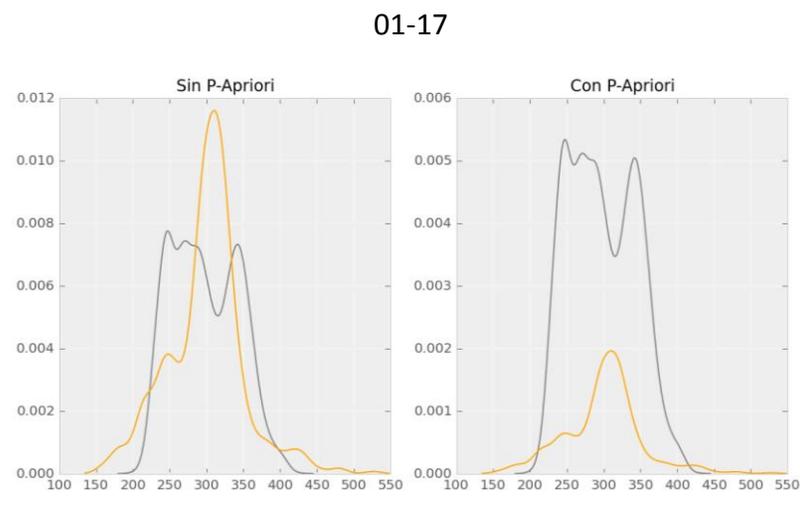
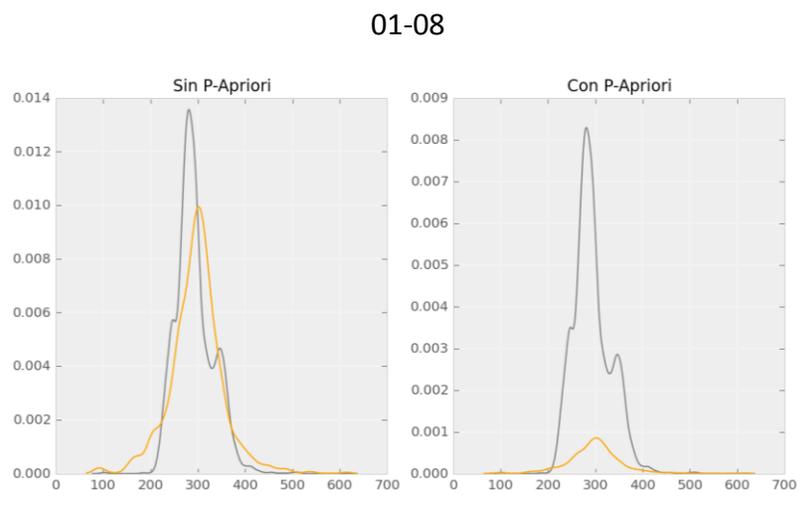
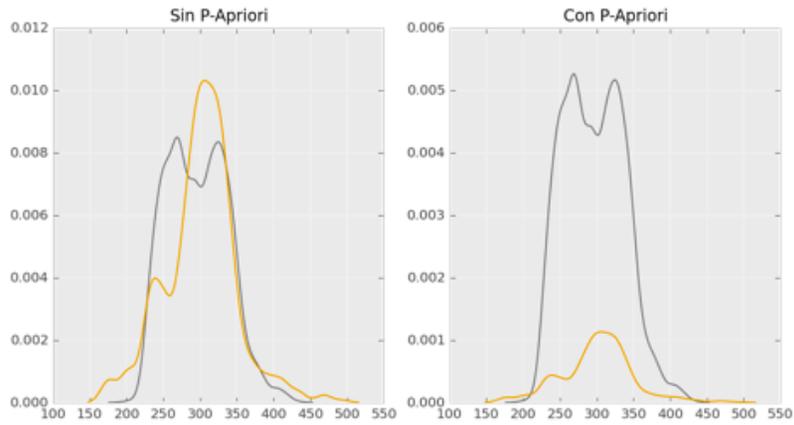
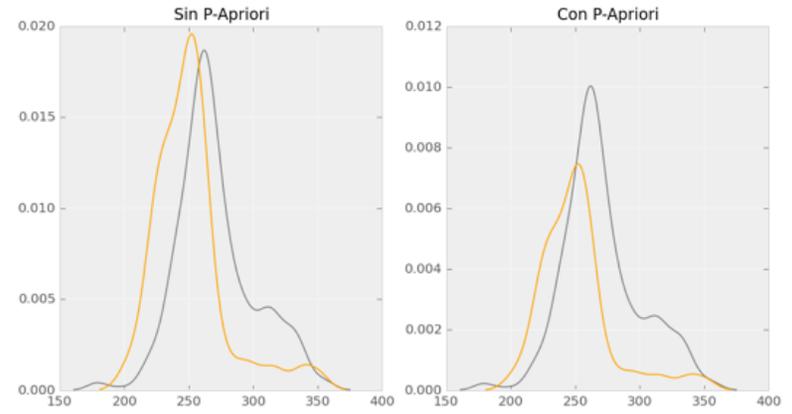


Figura 4.8: Curvas PDFs asociadas al registro sónico P para los pozos 01-08, 01-17, 05-16 y 08-08.

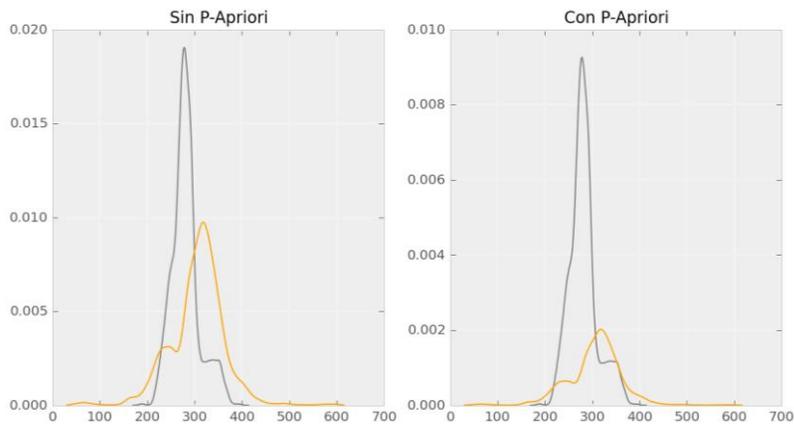
09-05



09-17



11-08



12-16

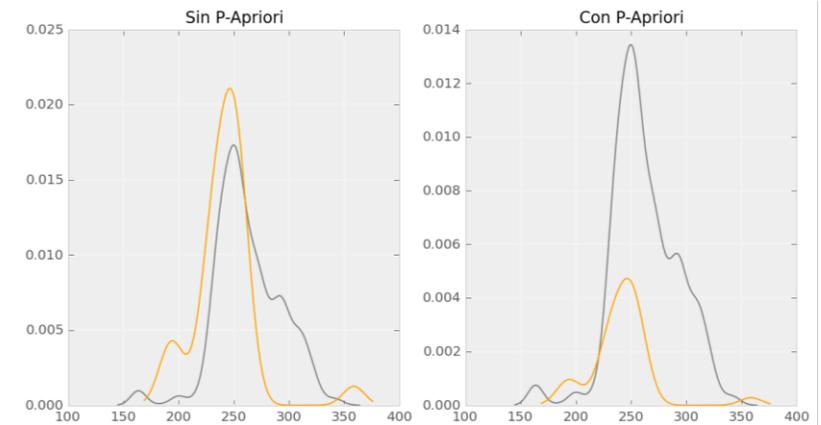
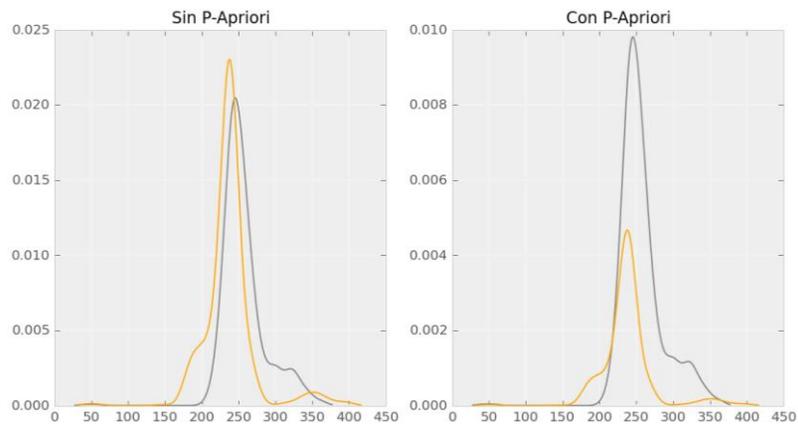
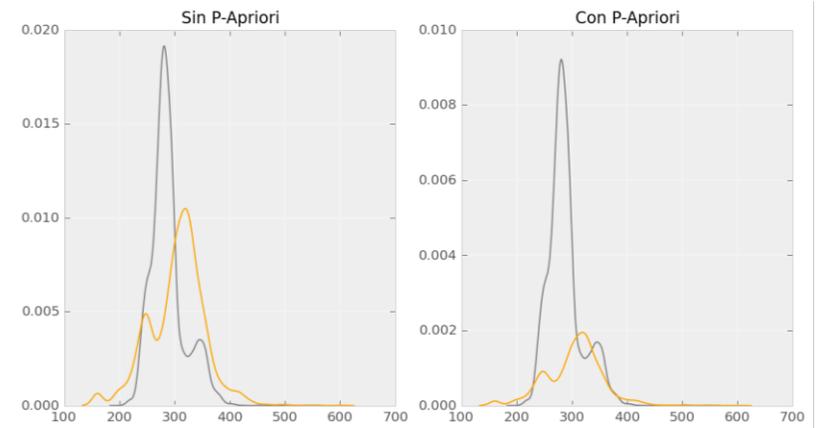


Figura 4.9: Curvas PDFs asociadas al registro sónico P para los pozos 09-05, 09-17, 11-08 y 12-16.

13-16



14-09



16-08

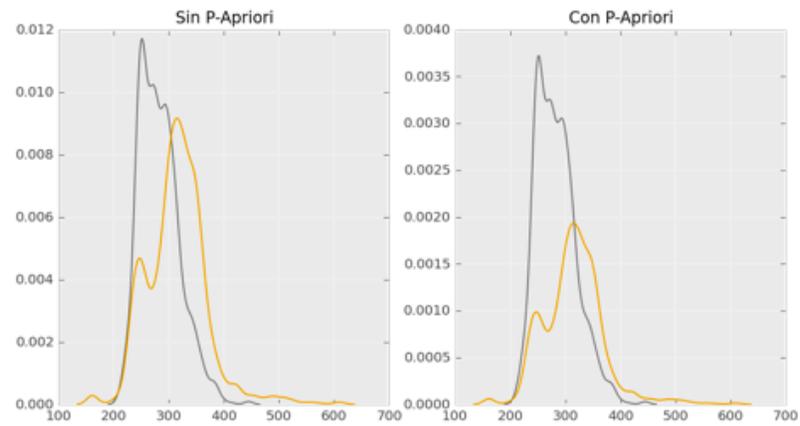


Figura 4.10: Curvas PDFs asociadas al registro sónico P para los pozos 13-16, 14-09 y 16-08.

La Tabla 4.1 representa el porcentaje de clasificación de la arena como tal derivadas de las matrices de decisión bayesiana obtenidas con las PDF presentadas previamente. La Figura 4.1 representa gráficamente los valores de esta tabla.

Primeramente se puede observar que los pozos 16-08, 08-08 y 01-08, pertenecientes al canal, presentaron una clasificación (con la probabilidad a priori) muy baja a excepción del segundo de ellos. De igual manera, los pozos 13-16 y 05-16 con los cuerpos arenosos identificados, también presentaron bajos valores de clasificación correcta. Por otro lado, el resto de los pozos, quienes presentaban intercalaciones de arenas y lutitas, presentaron una muy buena clasificación de las arenas con porcentajes entre 77% y 100%.

Pozos	Arena clasificada como arena	
	Sin la probabilidad a priori	Con la probabilidad a priori
<b>01-08</b>	0.54	0.28
<b>01-17</b>	0.55	1.00
<b>05-16</b>	0.74	0.27
<b>08-08</b>	0.63	0.92
<b>09-05</b>	0.61	0.77
<b>09-17</b>	0.66	0.83
<b>11-08</b>	0.53	0.48
<b>12-16</b>	0.85	0.93
<b>13-16</b>	0.70	0.20
<b>14-09</b>	0.61	0.79
<b>16-08</b>	0.41	0.30

Tabla 4.1: Porcentaje de clasificación correcto de la arena para el registro de sónico P, derivados de las matrices de decisión Bayesiana.

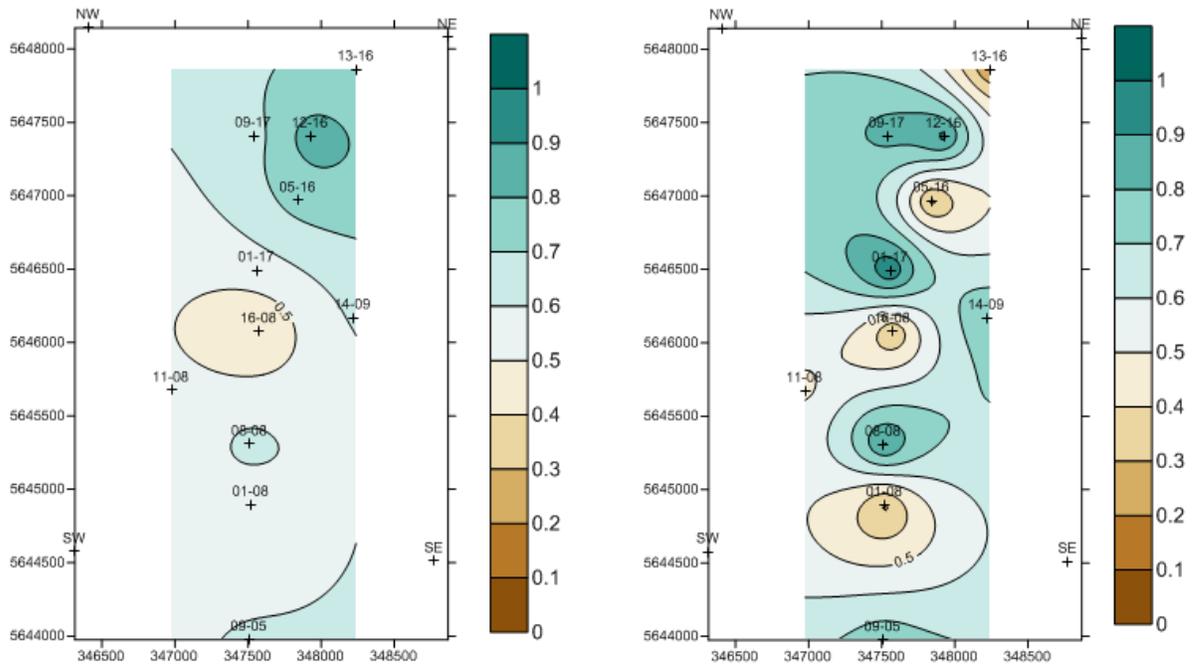
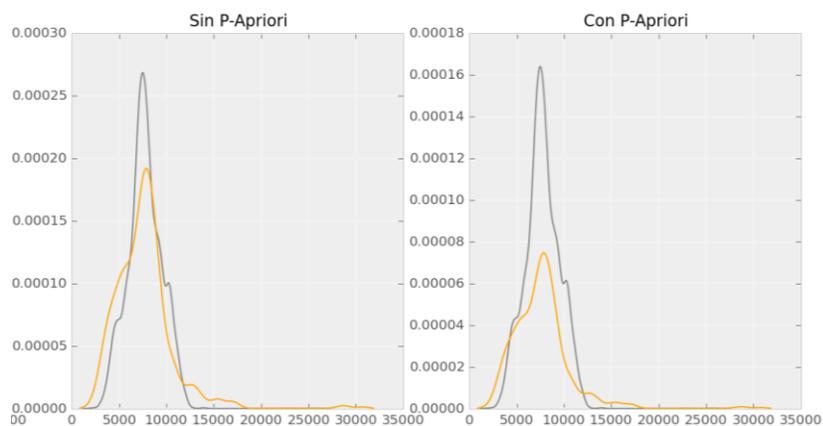


Figura 4.11: Arena clasificada como arena en el registro Sónico P. En el gráfico de la izquierda la probabilidad litológica a priori no está incluida pero sí en el gráfico de la derecha.

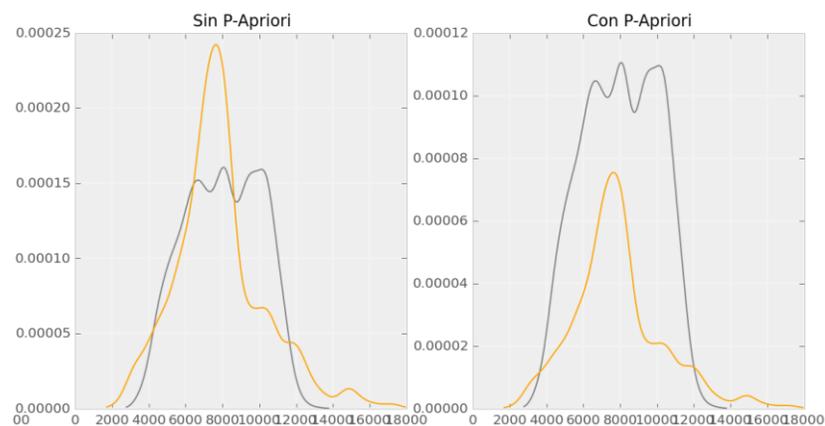
#### 4.5.1.2 Impedancia acústica

A diferencia de las PDFs del registro Sónico P, las PDFs del registro de impedancia acústica no fueron óptimas en todos los pozos. Estas curvas fueron descartadas de los Pozos 11-08 (Figura 4.13) y 13-16 (Figura 4.14) y no fue posible generarlas para el pozo 12-16 por no tener disponible el registro de densidad Volumétrica. También puede observarse de manera general que las curvas PDFs de las curvas asociadas a arenas y lutitas, no varían mucho entre sí al tomar en cuenta la probabilidad a priori de estas en cada pozo.

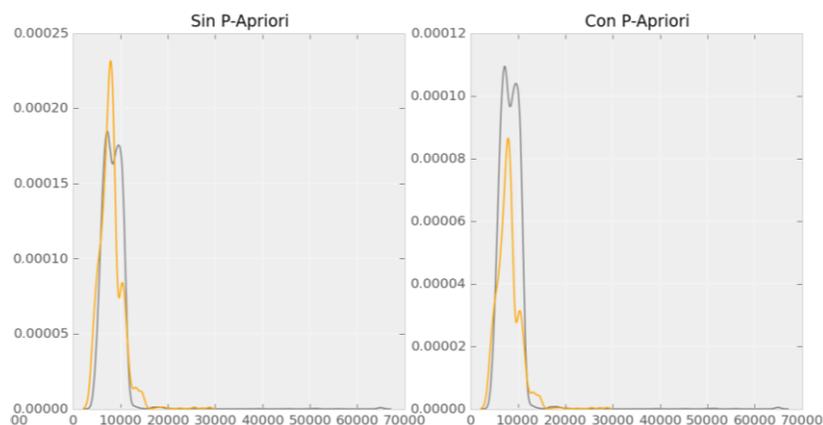
01-08



01-17



05-16



08-08

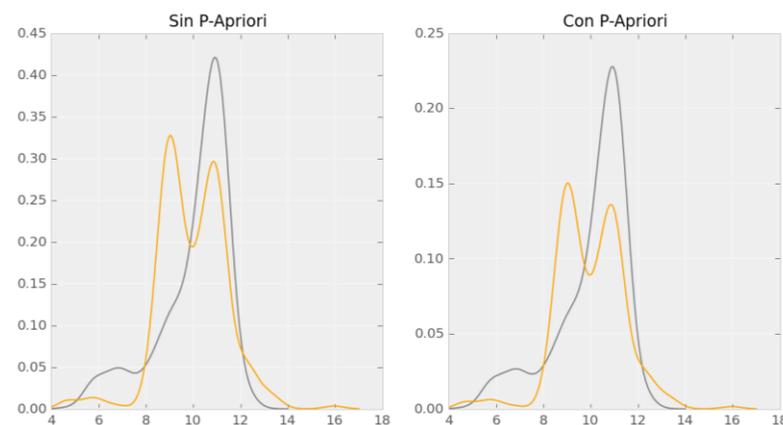
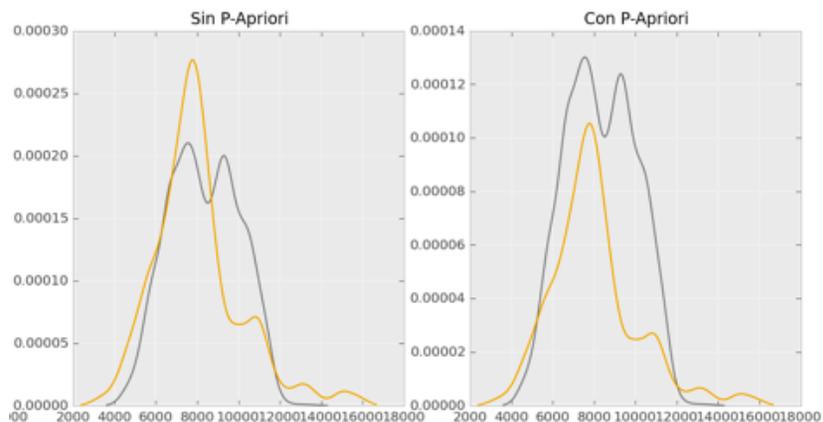
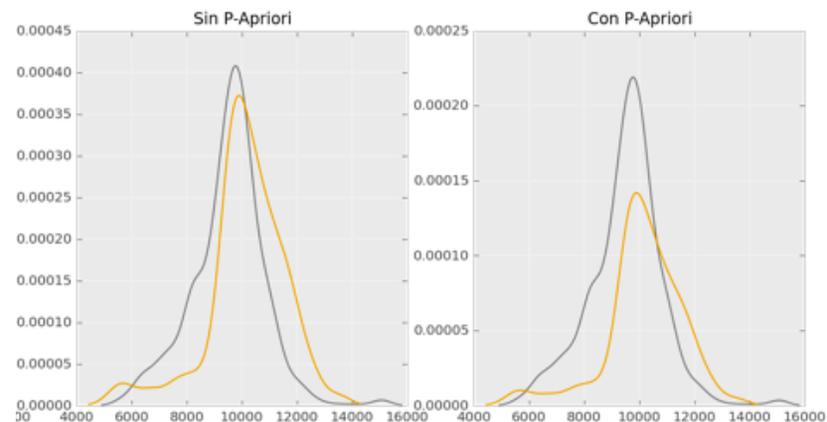


Figura 4.12: Curvas PDFs asociadas al registro de impedancia acústica para los pozos 01-08, 01-17, 05-16 y 08-08.

09-05



09-17



11-08

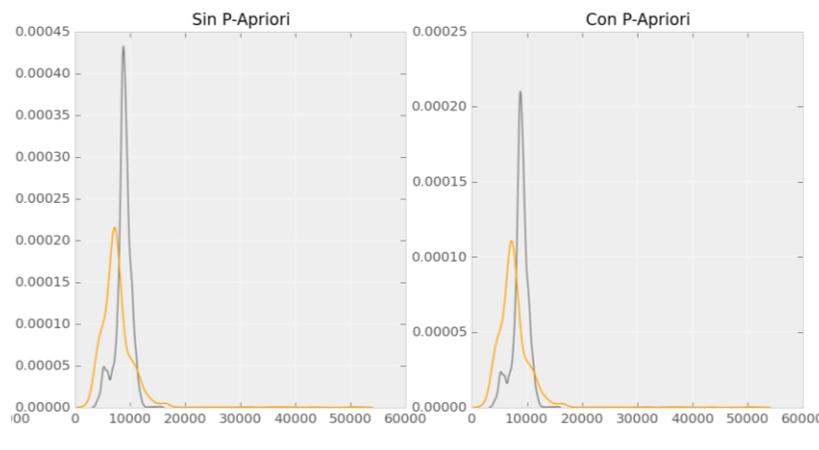
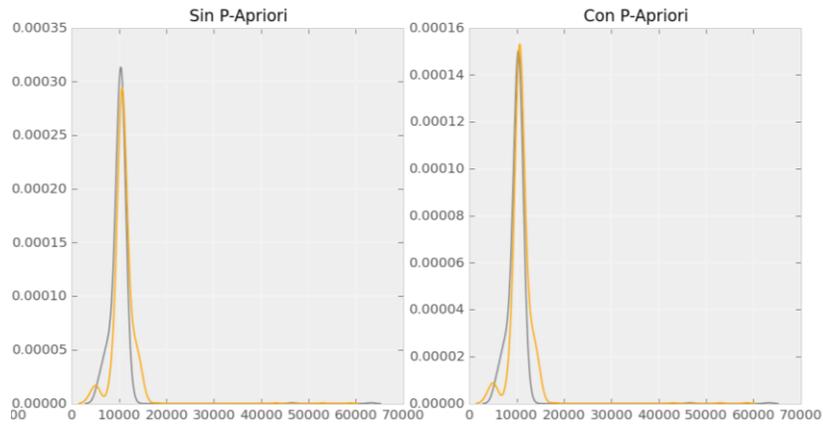
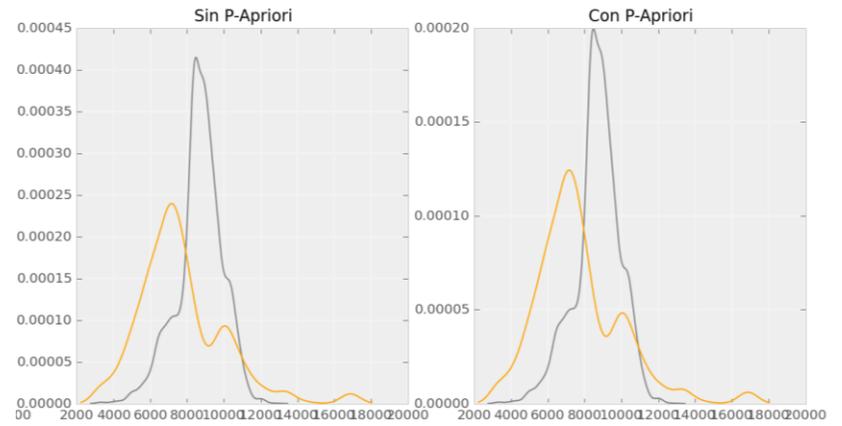


Figura 4.13: Curvas PDFs asociadas al registro de impedancia acústica para los pozos 09-05, 09-17, 11-08 y 12-16.

13-16



14-09



16-08

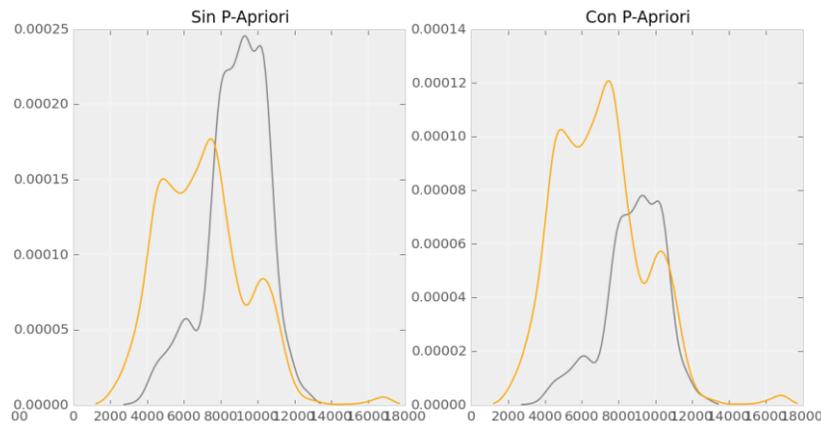


Figura 4.14: Curvas PDFs asociadas al registro de impedancia acústica para los pozos 13-16, 14-09 y 16-08.

La Tabla 4.2 representa el porcentaje de clasificación de la arena como tal derivadas de las matrices de decisión bayesiana obtenidas con las PDF presentadas previamente. La Figura 4.15 representa gráficamente los valores de esta tabla. Se puede evidenciar que existe una tendencia general a valores de clasificación (con la probabilidad a priori) relativamente altos, mayores a un 65%. Los pozos del canal, 16-08, 08-08 y 01-08, presentaron una casi perfecta clasificación de la arenas, a excepción del pozo productor 01-08 que presento un porcentaje del 68%. Un valor similar (65%) se evidenció para el pozo 05-16, el cual presenta un cuerpo arenoso. El resto de los pozos, con la característica intercalación de arenas y lutitas presentaron valores de discriminación superiores al 88%.

Pozos	Arena clasificada como arena	
	Sin la probabilidad a priori	Con la probabilidad a priori
12-16	n/a	n/a
09-17	0.54	0.88
01-08	0.58	0.68
01-17	0.52	1.00
05-16	0.76	0.65
08-08	n/a	1.00
09-05	0.58	1.00
11-08	0.45	n/a
13-16	n/a	n/a
14-09	0.52	1.00
16-08	0.45	0.99

Tabla 4.2: Matrices de decisión Bayesiana simplificada para el registro de impedancia acústica.

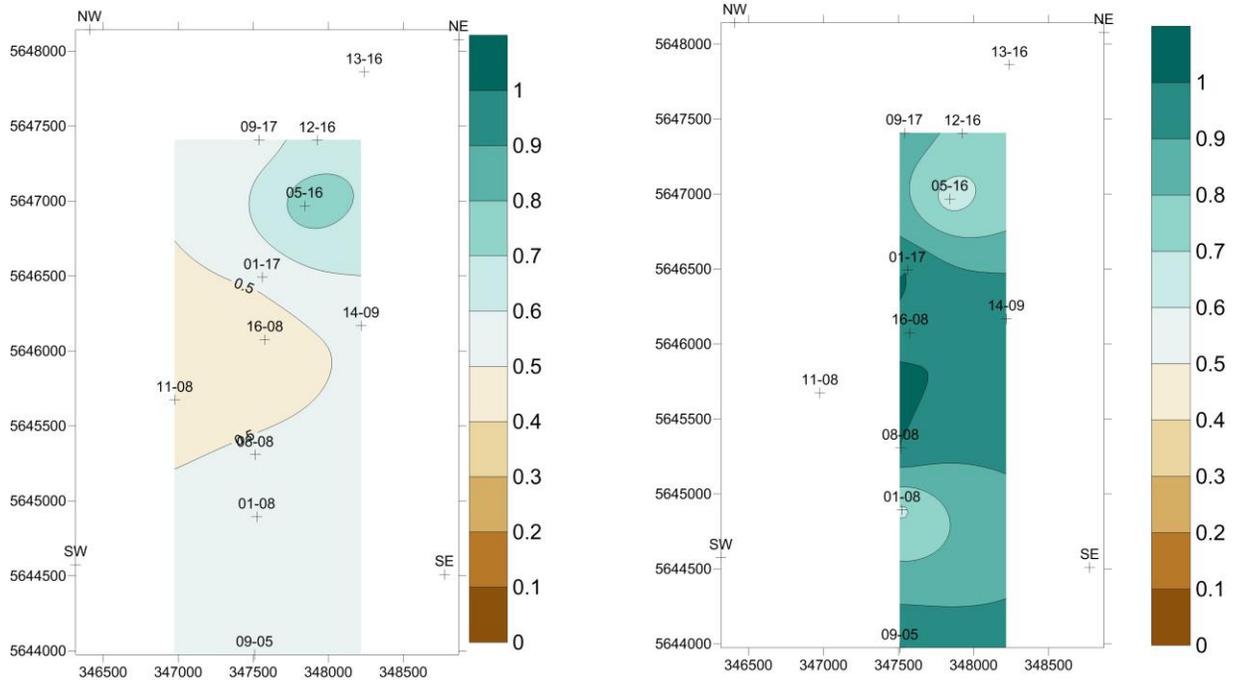
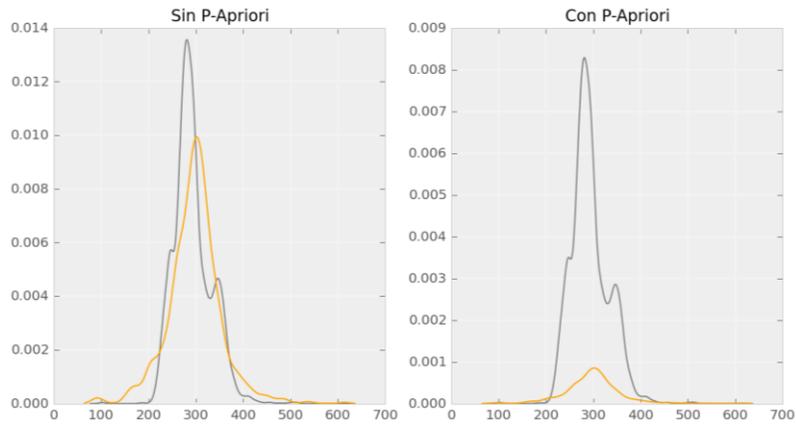


Figura 4.15: Arena clasificada como arena en el registro de impedancia acústica. En el gráfico de la izquierda la probabilidad litológica a priori no está incluida pero sí en el gráfico de la derecha.

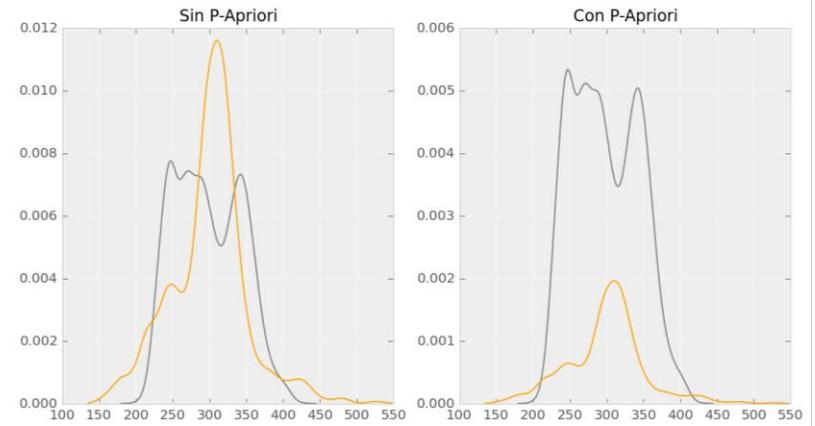
#### 4.5.1.3 Densidad volumétrica

El comportamiento de las PDFs del registro de Densidad Volumétrica (Figura 4.16, Figura 4.17 y Figura 4.18) reflejan los comportamientos característicos de los registros mencionados previamente. Por un lado, fueron descartadas las PDFs de los pozos 12-16, 09-17 y 13-16 por no presentar un comportamiento aceptable que permitiera una correcta clasificación. Y por otro lado, las probabilidades a priori generaron una diferencia entre las curvas de arenas y lutitas ocasionando variaciones considerables de los valores de probabilidad de discriminación.

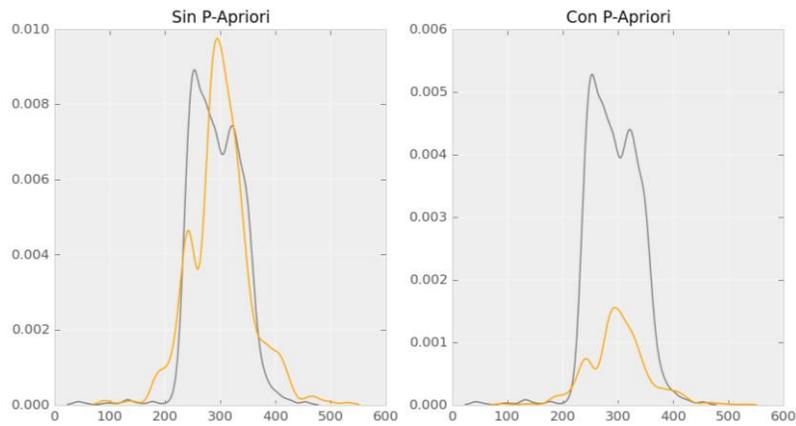
01-08



01-17



05-16



08-08

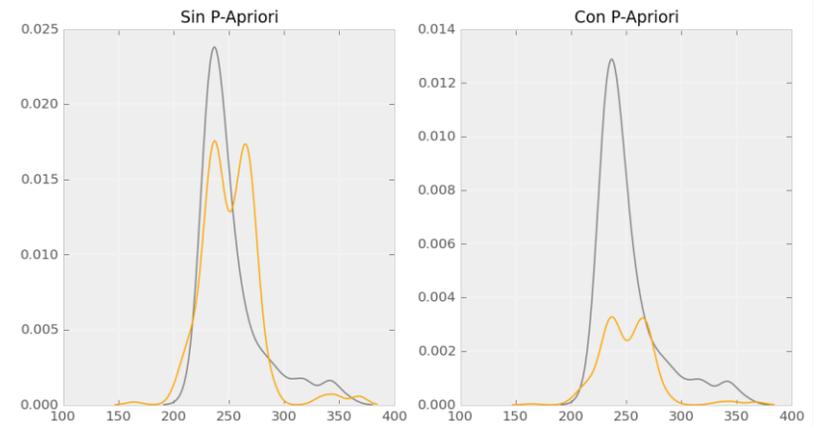
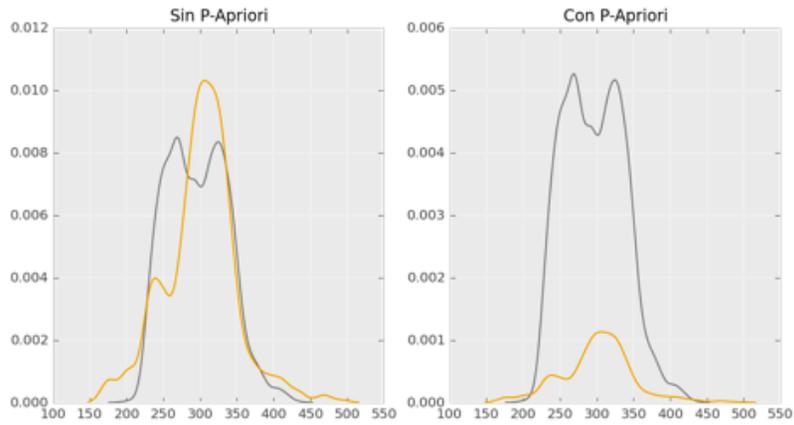
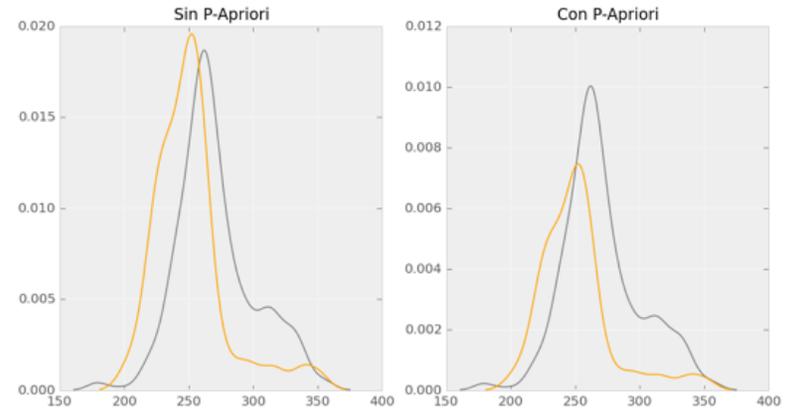


Figura 4.16: Curvas PDFs asociadas al registro de densidad volumétrica para los pozos 01-08, 01-17, 05-16 y 08-08.

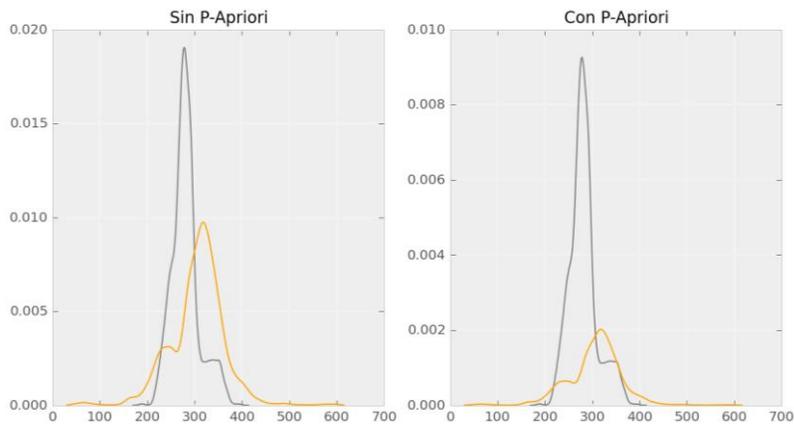
09-05



09-17



11-08



12-16

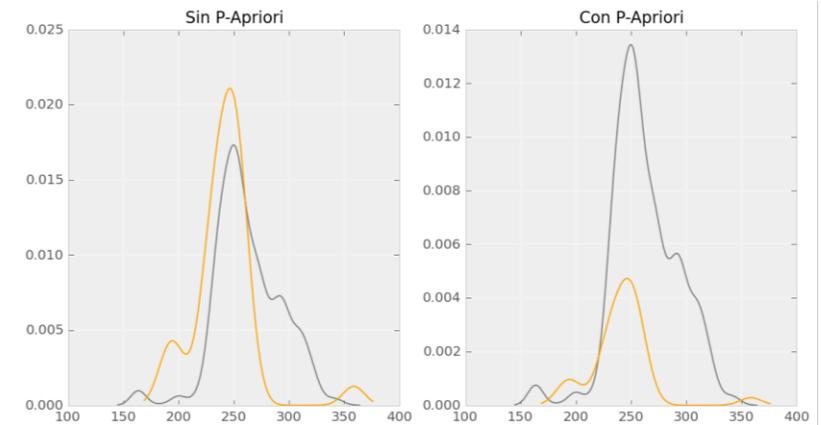
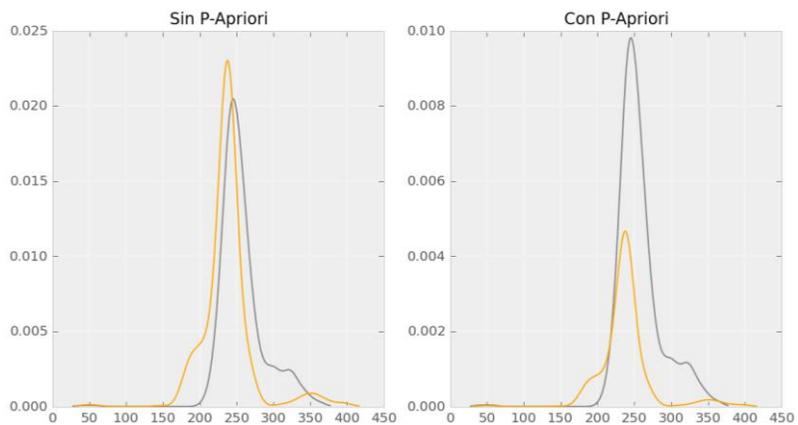
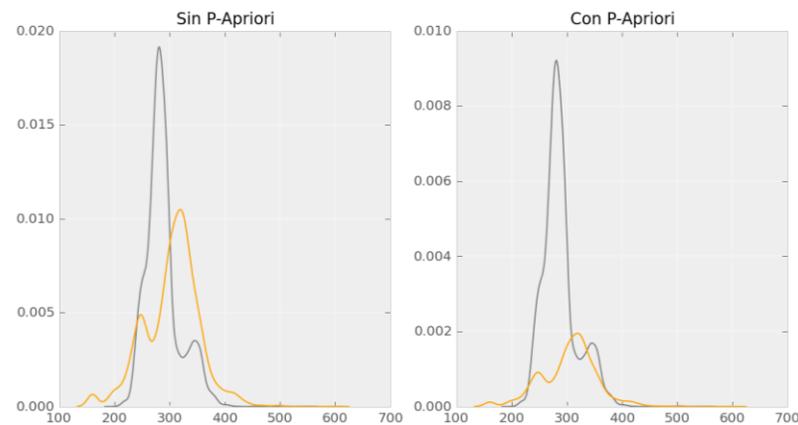


Figura 4.17: Curvas PDFs asociadas al registro de densidad volumétrica para los pozos 09-05, 09-17, 11-08 y 12-16.

13-16



14-09



16-08

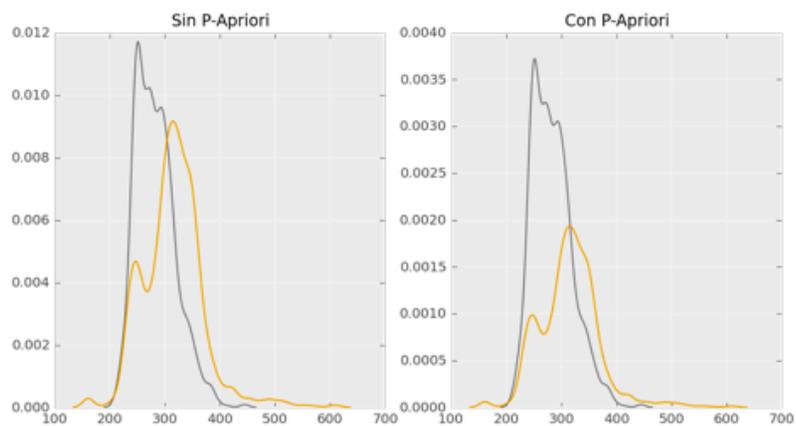


Figura 4.18: Curvas PDFs asociadas al registro de densidad volumétrica para los pozos 13-16, 14-09 y 16-08.

La Tabla 4.3 representa el porcentaje de clasificación de la arena como tal derivadas de las matrices de decisión bayesiana obtenidas con las PDF presentadas previamente. La Figura 4.19 representa gráficamente los valores de esta tabla.

El primer comportamiento evidenciado es que el porcentaje de clasificación es relativamente bajo, entre 43% y 79%. Dos de los pozos de canal presentaron valores altos de clasificación (60 y 69%) a excepción del pozo productor (01-08) un porcentaje del 52%. El resto de los pozos con intercalaciones de arenas y lutitas presentaron un porcentaje cercano al 50% a excepción del pozo 09-05 que presentó el porcentaje más alto de clasificación, 79%.

Pozos	Lutita clasificada como lutita	
	Sin la probabilidad a priori	Con la probabilidad a priori
<b>12-16</b>	n/a	n/a
<b>09-17</b>	n/a	n/a
<b>01-08</b>	0.69	0.52
<b>01-17</b>	0.83	0.49
<b>05-16</b>	0.72	0.49
<b>08-08</b>	0.47	0.68
<b>09-05</b>	0.34	0.79
<b>11-08</b>	0.71	0.42
<b>13-16</b>	n/a	n/a
<b>14-09</b>	0.76	0.43
<b>16-08</b>	0.78	0.60

Tabla 4.3: Matrices de decisión Bayesiana simplificada para el registro de densidad bulk.

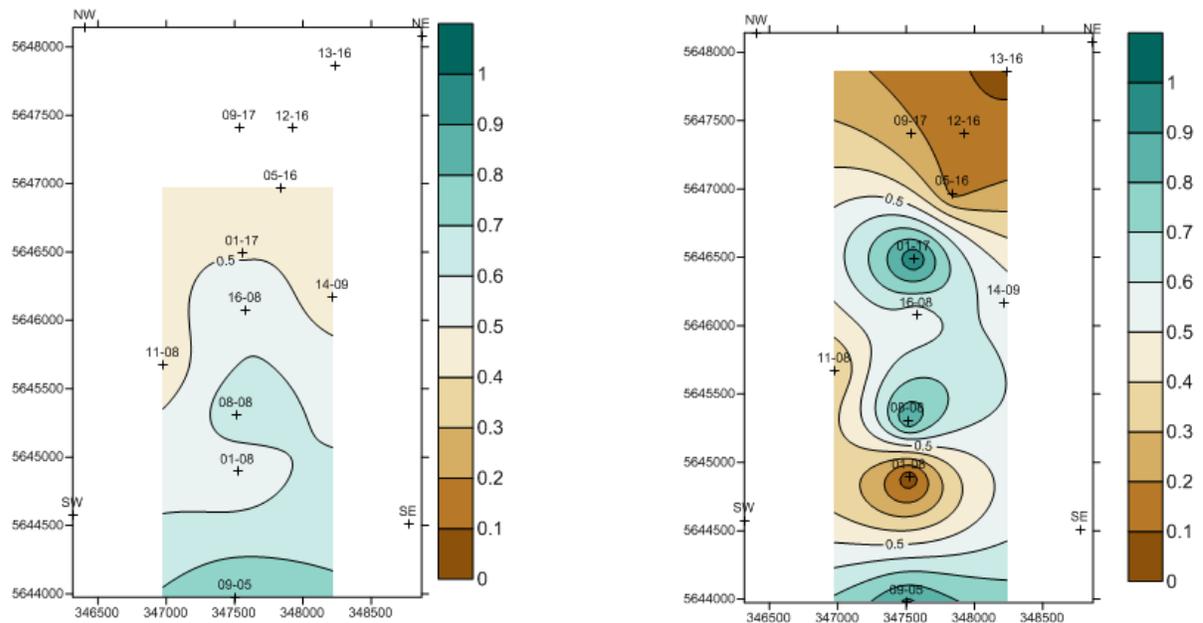


Figura 4.19: Arena clasificada como arena en el registro de densidad bulk. En el gráfico de la izquierda la probabilidad litológica a priori no está incluida pero sí en el gráfico de la derecha.

#### 4.5.2 Curvas PDFs bivariadas

Se generaron tres gráficos bivariados que reflejaban las posibles combinaciones sin repetición de los registros Sónico P, Impedancia Acústica y Densidad Volumétrica. Estos gráficos (Figura 4.20, Figura 4.21 y Figura 4.22) sin tomar en cuenta las probabilidades a priori, su porcentaje de clasificación variaba entre un 50% y 100%, pero al tomar en cuenta las probabilidades a priori de las arenas y lutitas se pudo evidenciar un aumento significativo y uniforme de estos porcentajes, llegando a un rango entre el 90% y 100% de acertación.

El único comportamiento destacable fue que el mapa Sónico P y Densidad Volumétrica (con la probabilidad a priori) fue que de los pozos pertenecientes al canal, presentaron valores cercanos al 100% de clasificación de la arenas, a excepción del pozo productor que presentó un porcentaje del 89%.

### Sónico P vs. Impedancia acústica

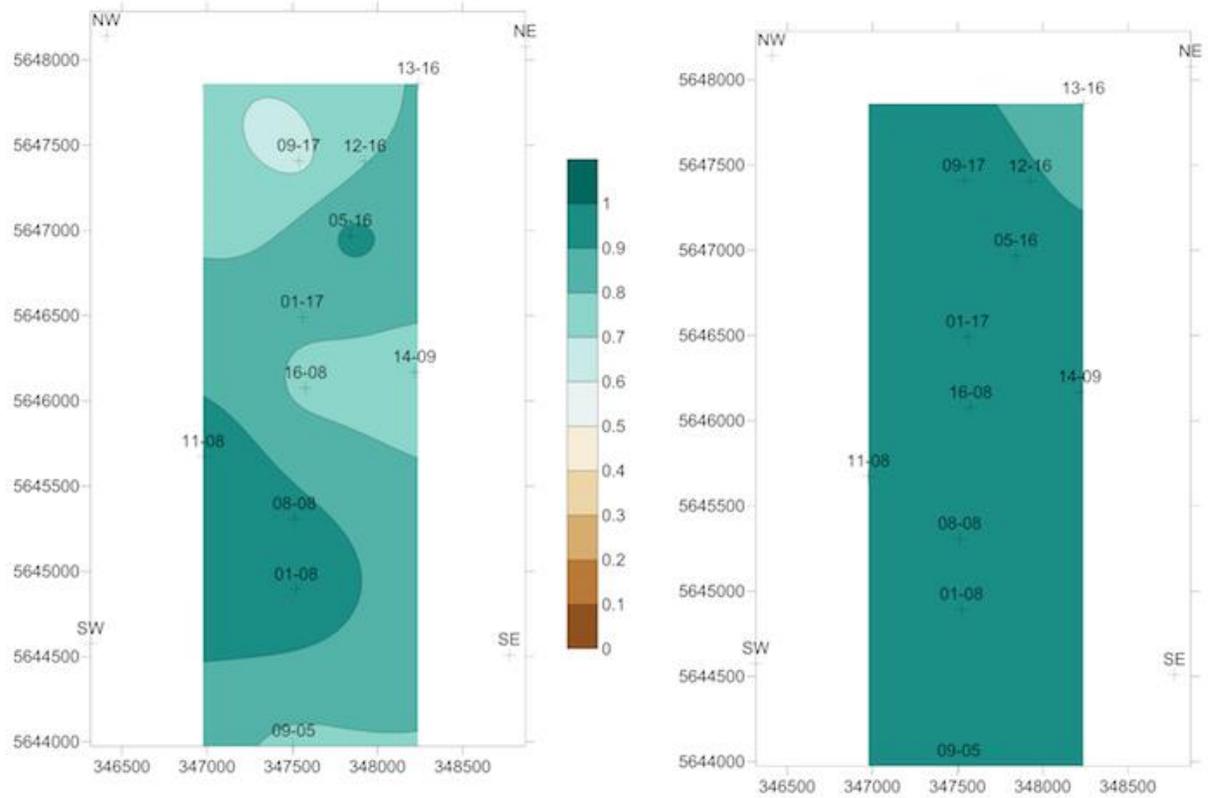


Figura 4.20: Representación gráfica de las Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Sónico P e Impedancia acústica.

### Sónico P vs. Densidad Volumétrica

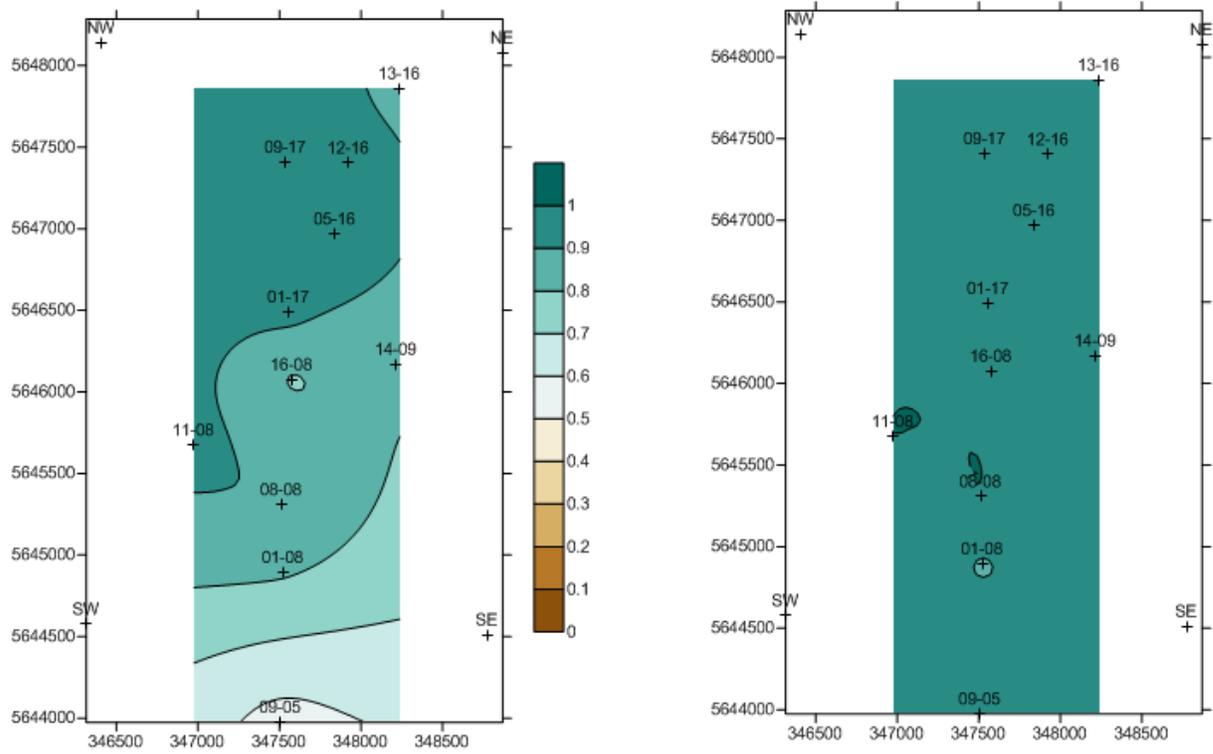


Figura 4.21: Representación gráfica de las Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Sónico P y Densidad Volumétrica.

## Impedancia acústica vs. Densidad Volumétrica

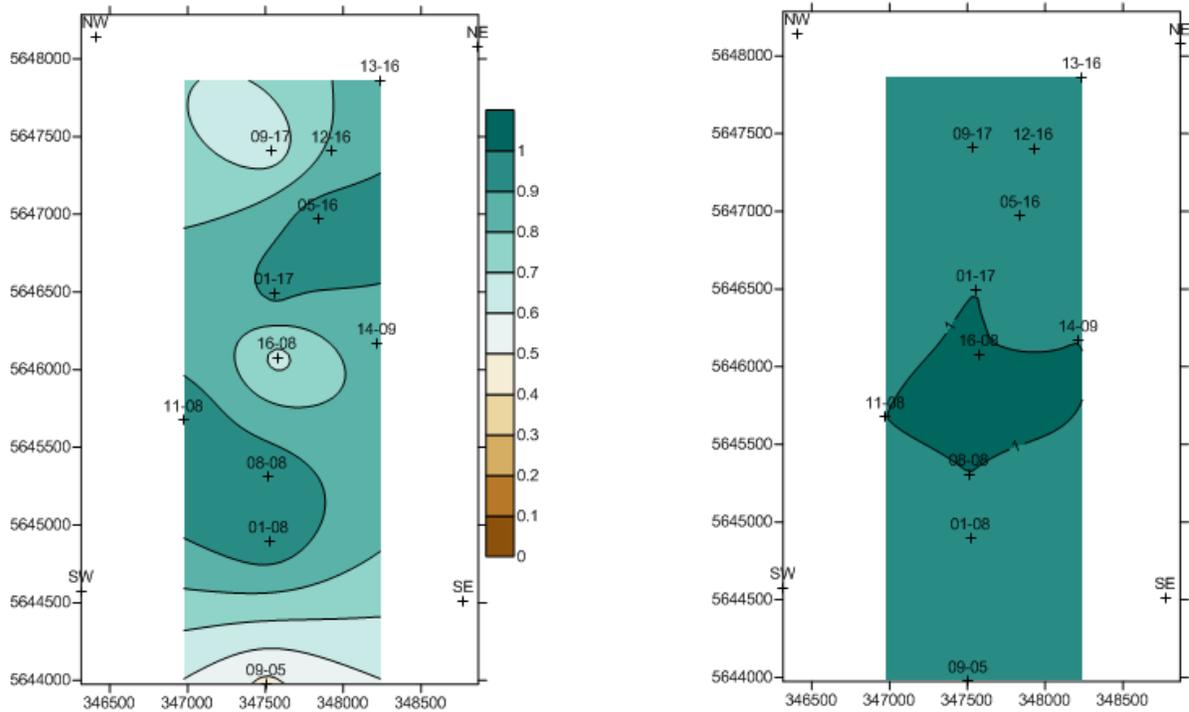


Figura 4.22: Representación gráfica de las Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Impedancia Acústica y Densidad Volumétrica.

### 4.6 Discriminación de fluidos

La interpretación petrofísica permitió identificar posibles zonas con gas y posibles acuíferos en ciertos pozos, la Tabla 4.4 resume esta información. En ella se expresa los fluidos presentes y cual de esos fluidos estaba siendo clasificado apropiadamente con las PDFs generadas para cada propiedad petrofísica. Sin tomar en cuenta las probabilidades a priori, en todos los pozos donde estaba la posible presencia de agua y gas, a excepción del pozos 08-08, el fluido que mejor clasificado era el agua, pero con respecto a todo los otros grupos (la Figura 4.23 y Figura 4.24 representa el comportamiento general de los pozos). Luego, tomando en cuenta las probabilidades

a priori, se evidencia que la discriminación por agua deja de ser notable y es superada por la curva de las lutitas (Figura 4.23 y Figura 4.24), se puede apreciar que el único grupo discriminado correctamente corresponde al grupo de las lutitas.

Pozo	Fluidos presentes	Fluido clasificado (sin probabilidad a priori)
<b>01-08</b>	Agua y Gas	Agua
<b>01-17</b>	Agua y Gas	Agua
<b>05-16</b>	Agua y Gas	Agua
<b>08-08</b>	Gas	Gas
<b>09-05</b>	Ninguno	
<b>09-17</b>	Ninguno	
<b>11-08</b>	Ninguno	
<b>12-16</b>	Ninguno	
<b>13-16</b>	Ninguno	
<b>14-09</b>	Agua y Gas	Agua
<b>16-08</b>	Agua y Gas	Agua

Tabla 4.4: Presencia de fluidos en los pozos del campo Blackfoot

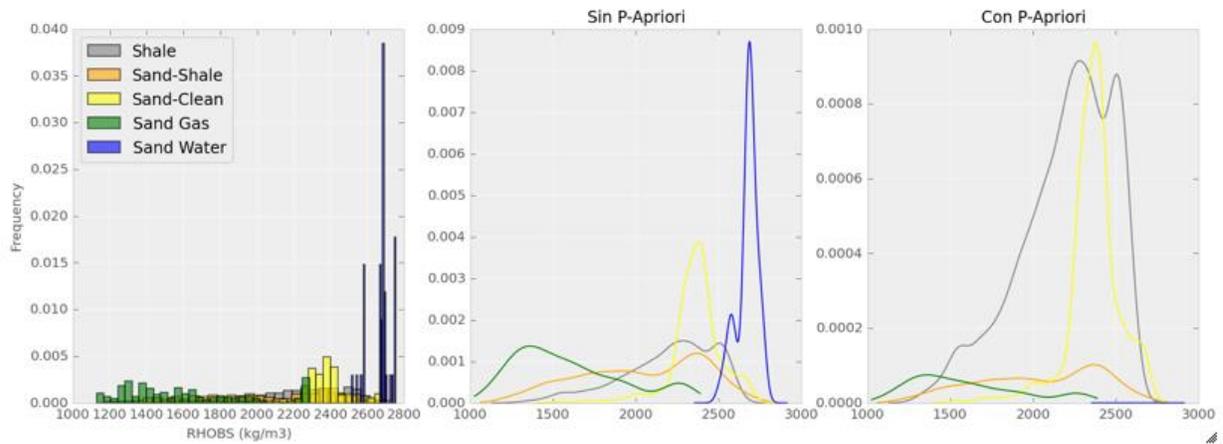


Figura 4.23: Curvas PDFs asociadas al registro de densidad del pozo 01-08.

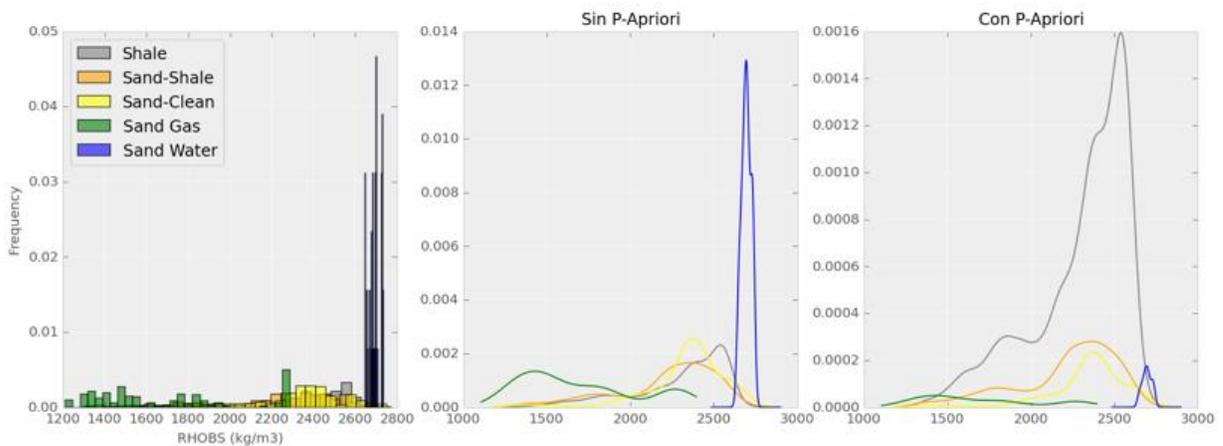


Figura 4.24: Curvas PDFs asociadas al registro de densidad del pozo 01-08.

#### 4.7 Discriminación de litología con atributos sísmicos

Con el software Petrel, fue generado una variedad de atributos: Impedancia acústica relativa, Coseno de la fase, Frecuencia dominante, Primera derivada de la frecuencia, Frecuencia instantánea, Fase instantánea, Amplitud original, Desfase en fase, Segunda derivada de la frecuencia, Amplitud RMS y Caos. Debido a que no se puede generar una evaluación litológica directa a los atributos sísmicos, siguiendo el proceso similar que se hace con la interpretación

petrofísica, todas las curvas PDF obtenidas provienen de la información de amplitud de cada atributo que atraviesa los diferentes pozos estudiados, teniendo en cuenta la evaluación litológica ya realizada en las evaluaciones petrofísicas previas.

#### 4.7.1 Curvas PDFs univariadas

De los 11 atributos generados (Tabla 4.5), 7 de ellos no discriminaron bien ninguna litología, el único atributo que puedo discriminar muy bien las arenas y lutitas fue el de Amplitud RMS y en el resto de los pozos se evidenció una discriminación parcial en algunos pozos. Los pozos pertenecientes al canal, 08-08 y 01-08, fueron discriminados por los atributos Frecuencia Dominantes y Amplitud RMS, el atributo Caos sólo pudo discriminar bien al pozo 08-08. Por otro lado, los estratos con lutitas y los cuerpos de arenas identificados en los pozos 13-16 y 05-16 se discriminaron bien con el atributo de Amplitud RMS y Caos.

Atributo	11-08	09-17	08-08	05-16	13-16	01-17	01-08	14-09
Impedancia acústica relativa	No							
Coseno de la fase	No							
Frecuencia dominante	No	Si	Si	No	No	No	Si	Si
Primera derivada de la frecuencia	No							
Frecuencia instantánea	Si	Si	Si	No	No	No	No	No
Fase instantánea	No	No	No	No	No	No	Si	No
Amplitud original	No							
Desfase en fase	No							
Segunda derivada de la frecuencia	No							

Amplitud RMS	Si	No	Si	Si	Si	No	Si	Si
Caos	No	No	Si	Si	Si	No	No	Si

Tabla 4.5: Discriminación litológica aceptable derivada de las PDF para los diferentes atributos sísmicos generados. Los recuadros en rojos representan una discriminación no válida.

Las curvas PDF que no discriminaron bien, presentaban un comportamiento similar a las PDF de la Figura 4.25. Se puede evidenciar que las curvas asociadas a cada litología presentan zonas que pueden asociarse a una buena discriminación, pero no son válidas ya que presentan una tendencia y un comportamiento similar.

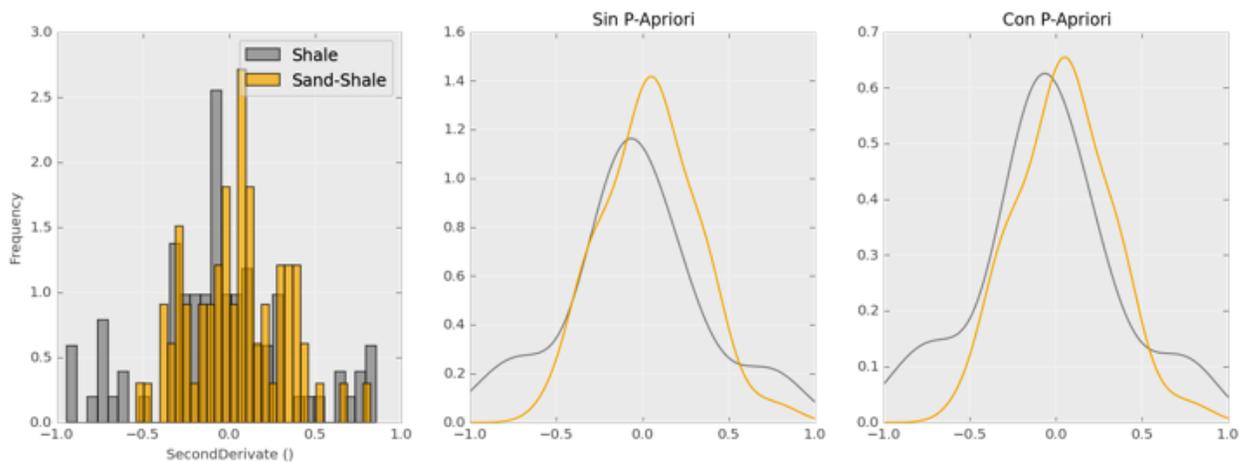


Figura 4.25: Curvas PDF pertenecientes al atributo de la Segunda Deriva del pozo 01-08.

#### 4.7.2 Curvas PDFs bivariadas

Fueron generados 55 gráficos bivariados, derivados de todas las posibles combinaciones resultantes de los 11 atributos generados. Las únicas combinaciones representativas fueron: Frecuencia Dominante vs Amplitud RMS, Frecuencia Dominante vs Amplitud RMS y Caos vs Amplitud RMS. Se evidencia un comportamiento similar al obtenido con la discriminación a través de la interpretación petrofísica, aquellos atributos que generaron una mejor clasificación con las PDF univariadas, fueron los que mejor clasificaron con las PDFs bivariadas.

De estas PDFs bivariadas se generaron diferentes matrices de decisión bayesiana, la Tabla 4.6 representa los porcentajes correctos de clasificación de arenas para las combinaciones de atributos mencionadas previamente. La primera evidencia destacable es que el aporte de la probabilidad a priori no generó, en las mayorías de los pozos, un cambio significativo de los porcentajes de clasificación. También puede apreciarse que la tendencia generalizada de los valores representas una muy buena clasificación de las arenas por encima de los 80% de veracidad. Los pozos asociados a los cuerpos de arenas identificados, 13-16 y 05-16, presentan porcentajes de discriminación similares lo cual podría darnos un indicio de que estos cuerpos podrían tener una composición mineralógica similar. Además, algo destacable es que sus porcentajes de clasificación eran unos de los más bajos en cada grupo de combinación de atributos. Un comportamiento similar se evidenciaba para los pozos pertenecientes al canal arenoso 01-08 y 08-08, siendo este último el que mejor valores de clasificación presentaba, coincidiendo con el pozos productor de la zona (Larsen, 1999).

Pozos	Frecuencia Dominante vs Caos		Frecuencia Dominante vs Amplitud RMS		Caos vs Amplitud RMS	
	A priori	A posteriori	A priori	A posteriori	A priori	A posteriori
01-08	0.82	0.82	0.97	0.96	0.82	0.82
01-17	0.72	0.72	0.90	0.85	0.90	0.90
05-16	0.89	0.70	0.99	0.77	0.88	0.69
13-16	0.85	0.73	0.90	0.86	0.91	0.76
08-08	0.95	0.95	0.88	0.95	0.98	0.93
09-17	0.82	0.82	0.84	0.84	0.78	0.78
11-08	0.71	0.97	0.90	0.96	0.76	0.95
14-09	1.00	1.00	0.97	1.00	0.90	0.99

Tabla 4.6: Porcentaje de clasificación correcto asociados a las arenas para tres tipos de combinación de atributos sísmicos.

### **4.7.3 Clasificación litológica de atributos sísmicos**

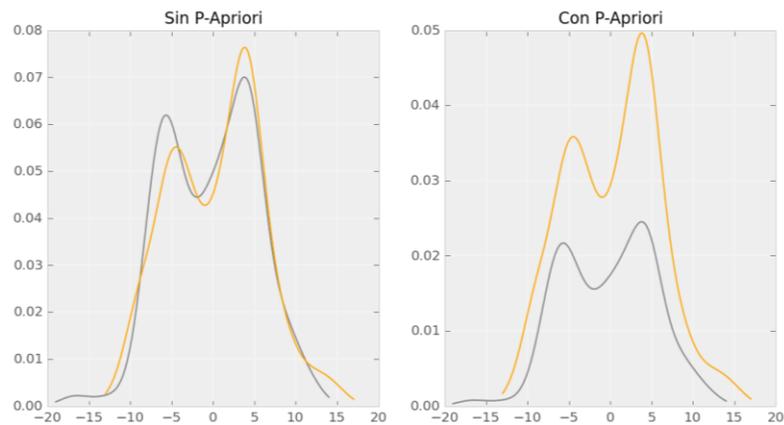
Los atributos sísmicos pueden obedecer a rasgos geológicos, estructurales o a presencia de fluidos, esto nos obligó a tomar las respuestas de cada atributo en cada pozo como un todo, ya que si queremos clasificar litológicamente todo la sísmica es necesario conocer el aporte de cada pozo identificados con la interpretación petrofísica, un método más confiable y directo.

Se generaron curvas PDF univariadas y bivariadas promedios creadas a partir de los resultados obtenidos en los dos apartados anteriores 4.7.1 (Curvas PDFs univariadas) y 4.7.2 (Curvas PDFs bivariadas), tomando como probabilidad a priori, de ocurrencia de arena un 70% y de lutita un 30%.

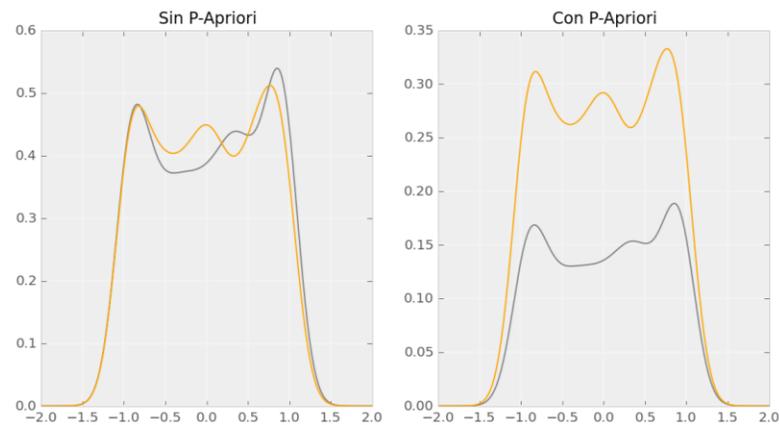
#### **4.7.3.1 Curvas PDFs univariadas promedios**

Las figuras Figura 4.26, Figura 4.27 y Figura 4.28 muestran las PDF univariadas promedio asociadas a cada atributo sísmico evaluado. Se puede evidenciar que la probabilidad priori hizo que la curva asociada a arenas sobresaliera sobre la curva de lutitas, ocasionando que en la mayoría de los atributos se tuvieron un porcentaje de aceptación de casi el 100%, valor que no se puede tomar como cierto porque las curvas PDF presentan una tendencia muy similar que no permiten hacer una buena discriminación litológica, no se permite establecer un porcentaje de discriminación aceptable del contenido de lutitas ya que este en promedio es casi nulo, comportamiento que contradice por completo las intercalaciones de arenas y lutitas características de la zona.

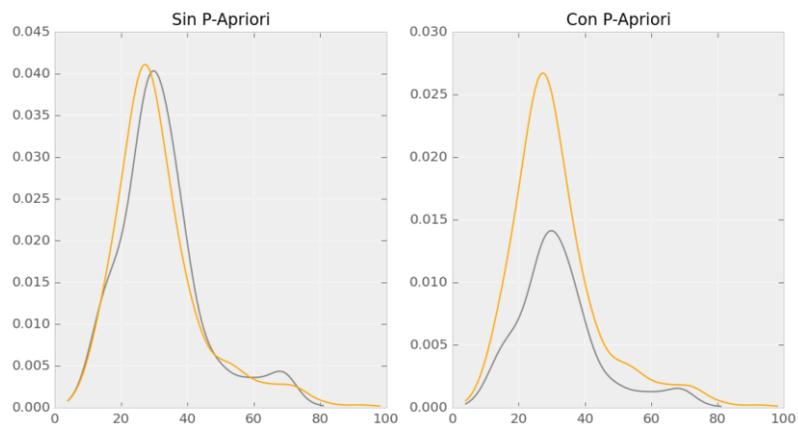
Impedancia acústica relativa



Coseno de la fase



Frecuencia dominante



Primera derivada

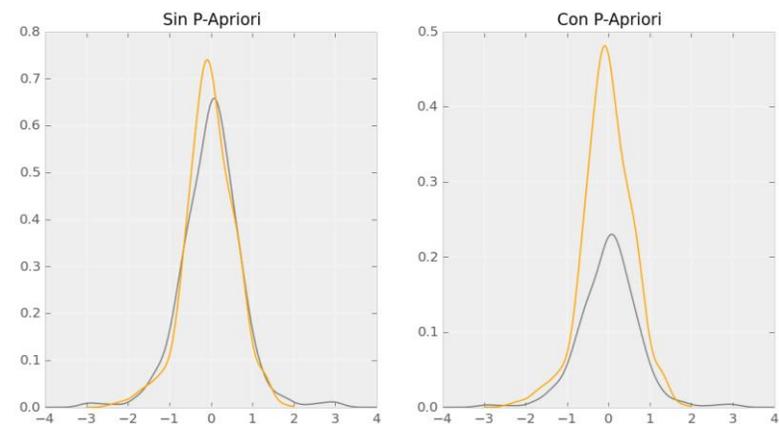
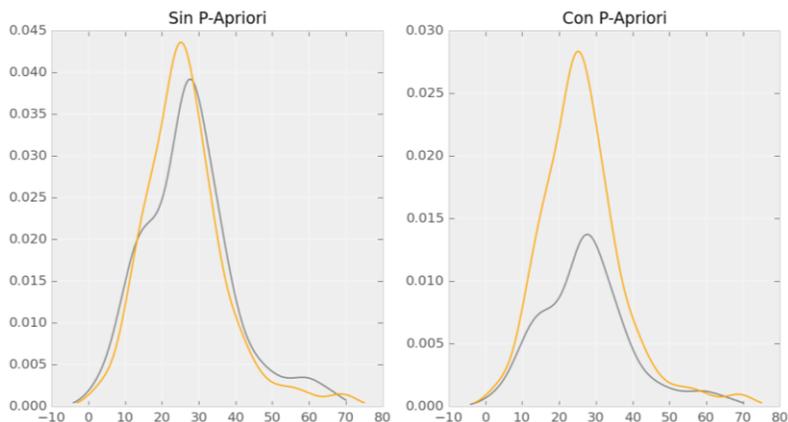
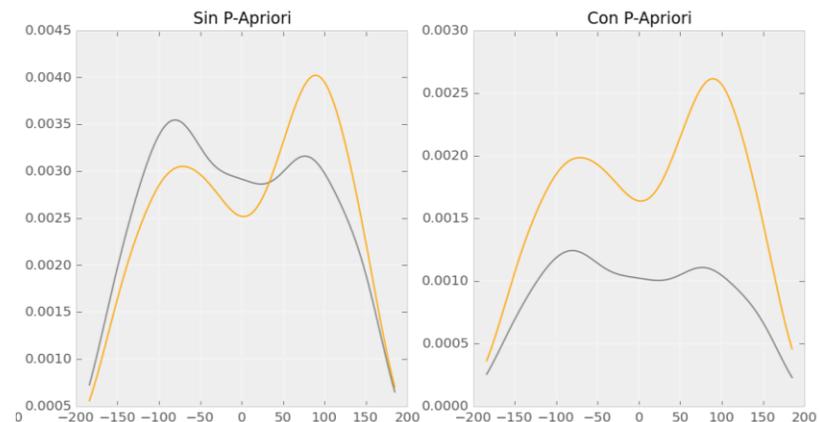


Figura 4.26: Curvas PDF promedio asociadas a cuatro atributos sísmicos distintos: Impedancia acústica Relativa, Coseno de la fase, Frecuencia Dominante y Primera Derivada.

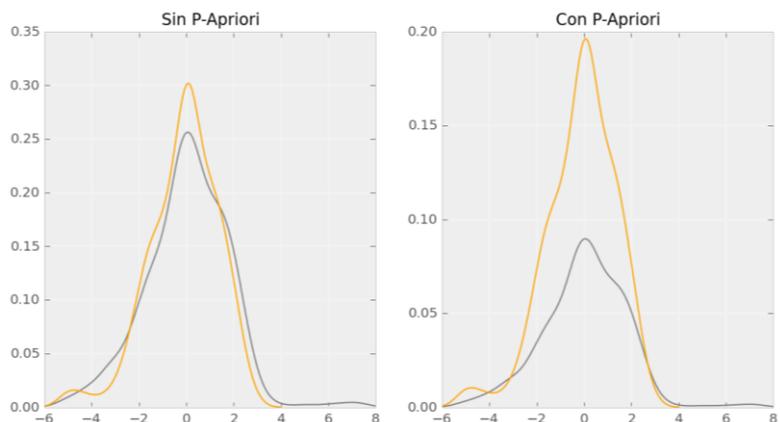
Frecuencia instantánea



Fase instantánea



Amplitud original



Desfase en fase

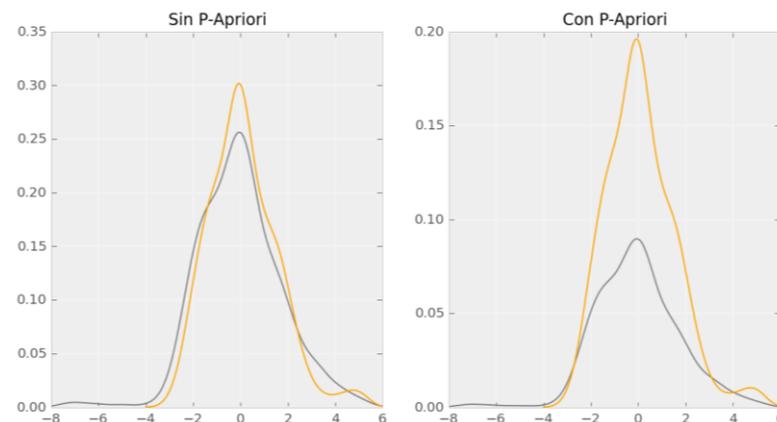
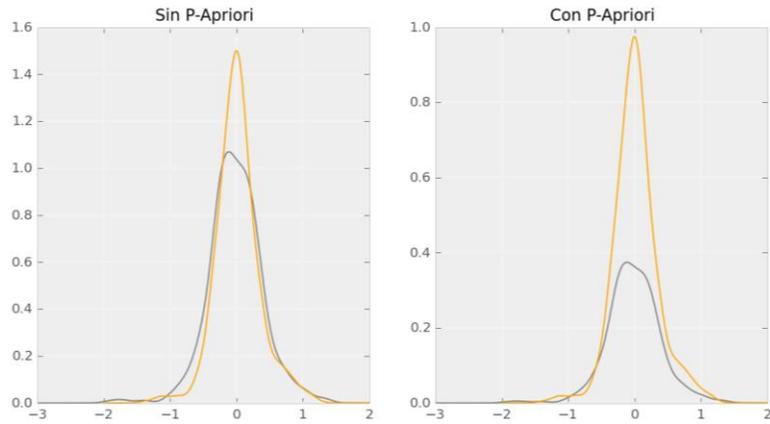
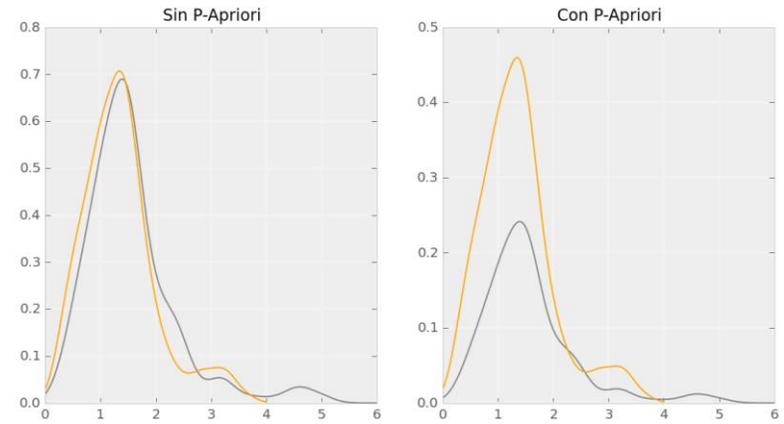


Figura 4.27: Curvas PDF promedios asociadas a cuatro atributos sísmicos distintos: Frecuencia instantánea, Fase instantánea, Amplitud original y Desfase en fase.

### Segunda derivada



### Amplitud RMS



### Caos

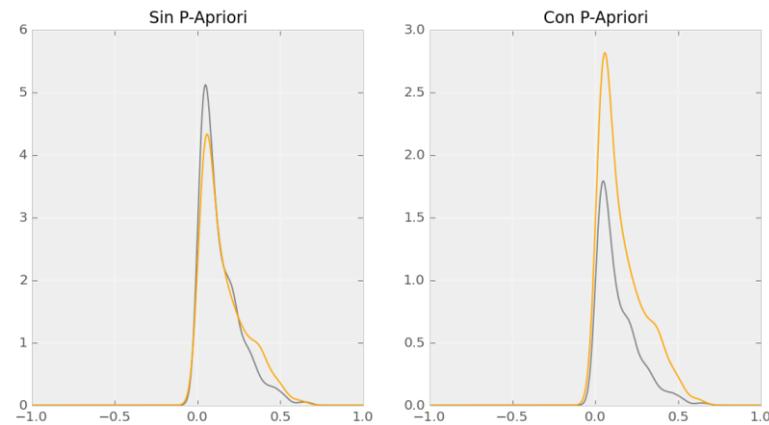


Figura 4.28: Curvas PDF promedios asociadas a cuatro atributos sísmicos distintos: Segunda derivada, Amplitud RMS, Amplitud original y

Caos.

### 4.7.3.2 Curvas PDFs bivariadas promedios

Se realizó una selección cuidadosa de las 55 gráficos bivariados promedios descartando aquellos gráficos que no mostraran una zonificación uniforme de los grupos litológicos en estudio. Por ejemplo, la gran mayoría de las curvas bivariadas tenían un comportamiento similar a la mostrada en la Figura 4.29, no se pueden identificar ninguna zonificación relativamente aisladas asociadas a las litologías presentes ni tampoco tendencias tangibles, se evidencia el mismo caso expuesto en el apartado anterior: las curvas PDF presentan un comportamiento similar en zonas compartidas casi en su totalidad.

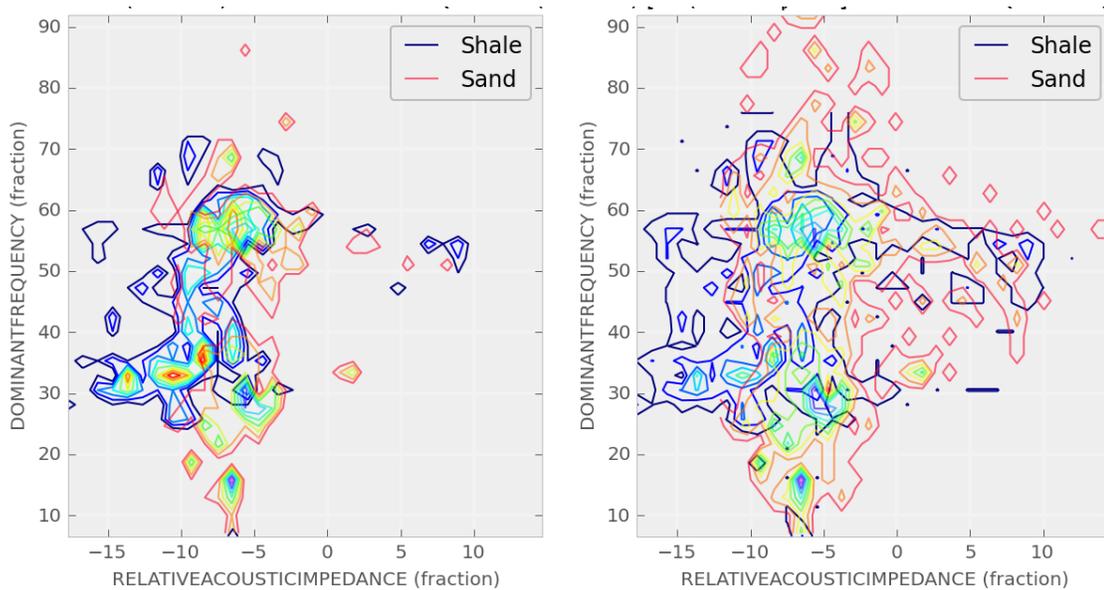


Figura 4.29: PDFs bivariadas para la combinación de atributos Impedancia Acústica Relativa vs Frecuencia Dominante. El gráfico de la derecha tiene el aporte de la probabilidad a priori litológica.

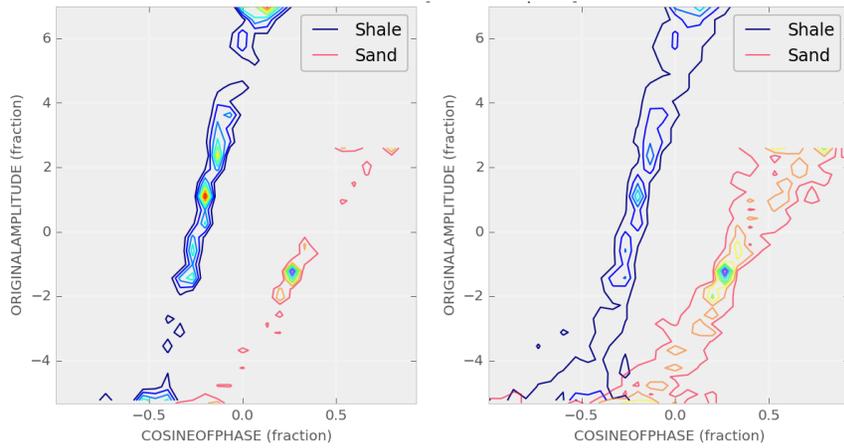
Los gráficos que cumplían nuestro proceso de selección fueron aquellos expuestos en la Tabla 4.7. Se evidencia, que existe un porcentaje de clasificación de las arenas muy altos, entre 85% 96% y que no existe una diferencia sustancial de los resultados al tomar en cuenta la probabilidad a priori de las arenas y lutitas.

Atributo sísmico	A priori	A posteriori
Coseno de la fase vs Amplitud Original	0.94	0.96
Frecuencia Dominante vs Frecuencia Instantánea	0.89	0.91
Desfase en la Fase vs Segunda Derivada	0.85	0.85
Desfase en la Fase vs Amplitud RMS	0.95	0.95

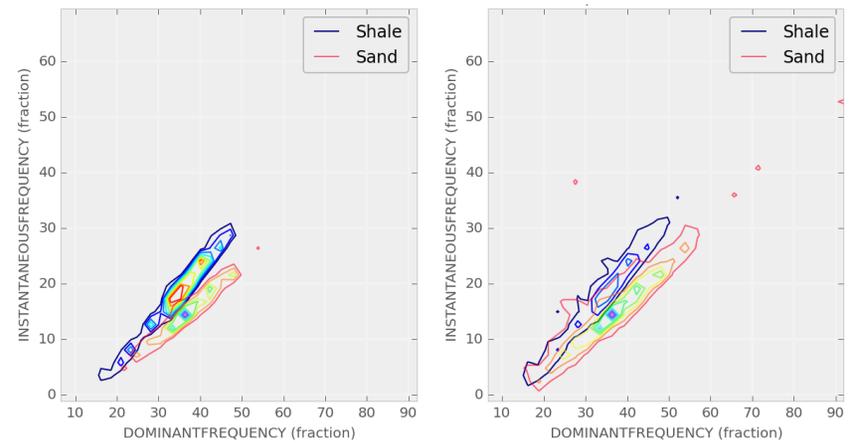
Tabla 4.7: Combinaciones de Atributos sísmicos que representaban zonificaciones correctas de los grupos litológicos en las PDF bivariadas correspondientes.

En la Figura 4.30 se observan las PDFs bivariadas correspondientes a la combinación de atributos de la Tabla 4.7. Se observa una tendencia generalizada de que las arenas se ubican en la parte inferior derecha, asociadas a tres de los pozos, a valores altos de los atributos que involucren la información de la fase y valores bajos del otro atributo en comparación. También se puede evidenciar una buena agrupación de los grupos litológicos, incluso en la combinación Frecuencia Dominante vs Frecuencia Instantánea, en donde los grupos presentan valores similares pero están muy bien delimitados.

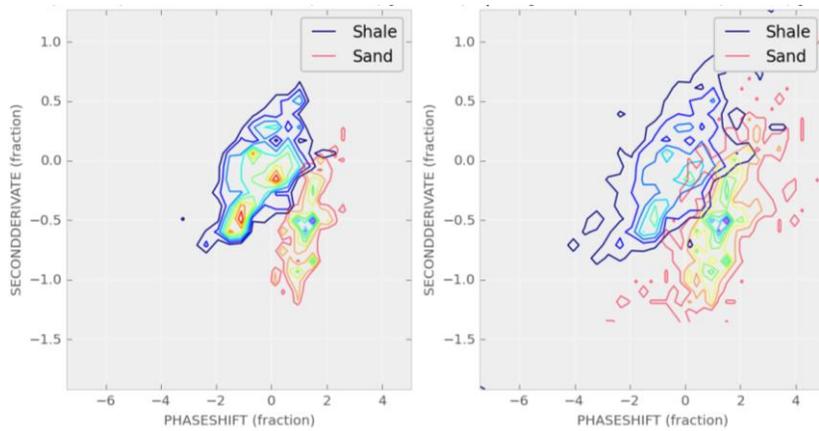
a) Coseno de la fase vs Amplitud Original



b) Frecuencia Dominante vs Frecuencia Instantánea



c) Desfase en la Fase vs Segunda Derivada



Desfase en la Fase vs Amplitud RMS

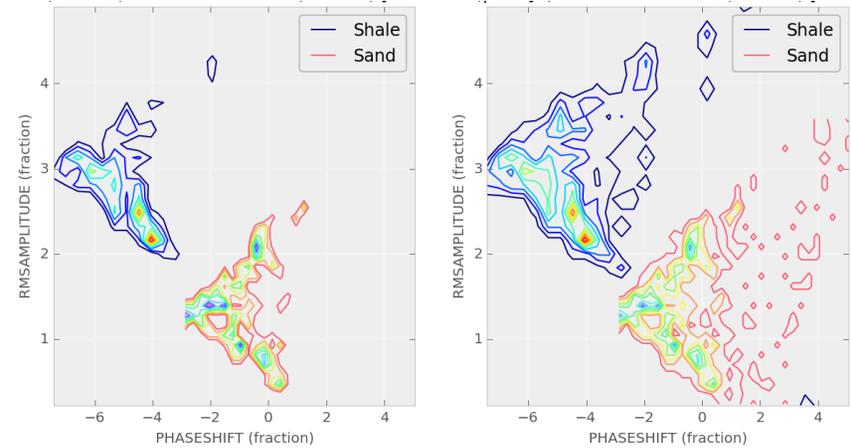


Figura 4.30: Curvas PDFs bivariadas para la combinación

#### **4.7.3.3 Mapa de clasificación litológica derivados de las Curvas PDFs bivariadas promedios**

Usa el procedimiento expuesto en la Metodología (página 43, Generación de mapas de probabilidad usando los atributos sísmicos), de las 5 combinaciones de atributos de la Tabla 4.7, sólo dos de ellas generaron mapas con tendencias identificables: Frecuencia Dominante vs. Frecuencia Instantánea (Figura 4.31b) e Impedancia Acústica Relativa vs Primera Derivada (Figura 4.32b).

Tal como se explicó en el marco teórico, estos mapas generados representan el porcentaje de arena identificado en cada zona de la sísmica, ya que se intentó clasificar cada par de amplitud en cada profundidad en el intervalo de estudio. Es por esto, que tiene más sentido comparar estos mapas con el mapa de contenido de arena (probabilidad a priori) generado a partir de la interpretación petrofísica.

Observando el mapa correspondiente a la combinación Frecuencia Dominante vs. Frecuencia Instantánea (Figura 4.31b) podemos constatar que la tendencia general está asociada a un bajo contenido de arena, hecho que contradice los resultados obtenidos con la interpretación petrofísica (Figura 4.31a). El pozo 01-17, con el mayor contenido de arena, presentó aproximadamente un 8% de ocurrencia de estratos arenosos. Los pozos pertenecientes al canal, 16-08, 08-08 y 01-08, que presentaban un contenido de arena superior, presentaron valores de ocurrencias de arenas entre 7% y 22, siendo el pozo productor 01-08 quien presentaba el valor más alto entre los pozos del canal. El pozo 13-16, con el posible cuerpo arenoso, conservó su porcentaje de contenido de arena, 51% con la interpretación petrofísica y un 52% con el mapa de la combinación de atributos sísmico en cuestión. Por otro lado, el otro pozo con el cuerpo arenoso, 05-17, presentó una diferencia grande entre los porcentajes de contenido de arenas, 43% asociado a la interpretación petrofísica y un 13% con el análisis de atributos. El resto de los pozos presentaron una diferencia en promedio entre los mapas del 55%.

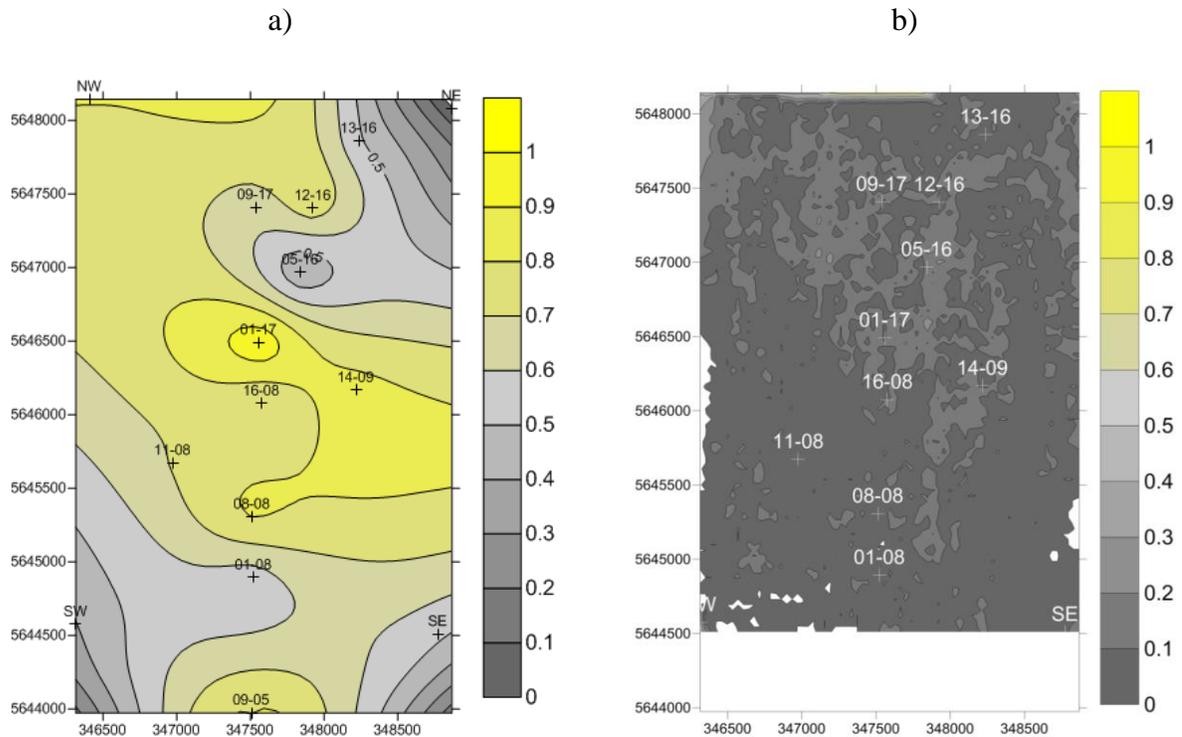


Figura 4.31: a) Mapa de probabilidad a priori (contenido de arena) derivado de la interpretación petrofísica b) Mapa de contenido de arena derivado de la clasificación por PDF bivariadas de atributos sísmicos correspondientes s la combinación Frecuencia Dominante vs. Frecuencia Instantánea.

Para la segunda combinación de atributos, Impedancia Acústica Relativa vs Primera Derivada (Figura 4.32b), se observa una tendencia similar: bajo contenido de arena. Los pozos del canal, 16-08, 08-08 y 01-08, presentaron un porcentaje de arena en promedio del 27%, con una diferencia del 41% en comparación a la interpretación petrofísica. Con respecto a los pozos asociados a los canales, 13-16 y 05-16, el primero de ellos presentó una diferencia del 32% y el segundo del 13%. El resto de los pozos presentó una diferencia en promedio del 46%, similar al porcentaje del mapa de la Figura 4.31.

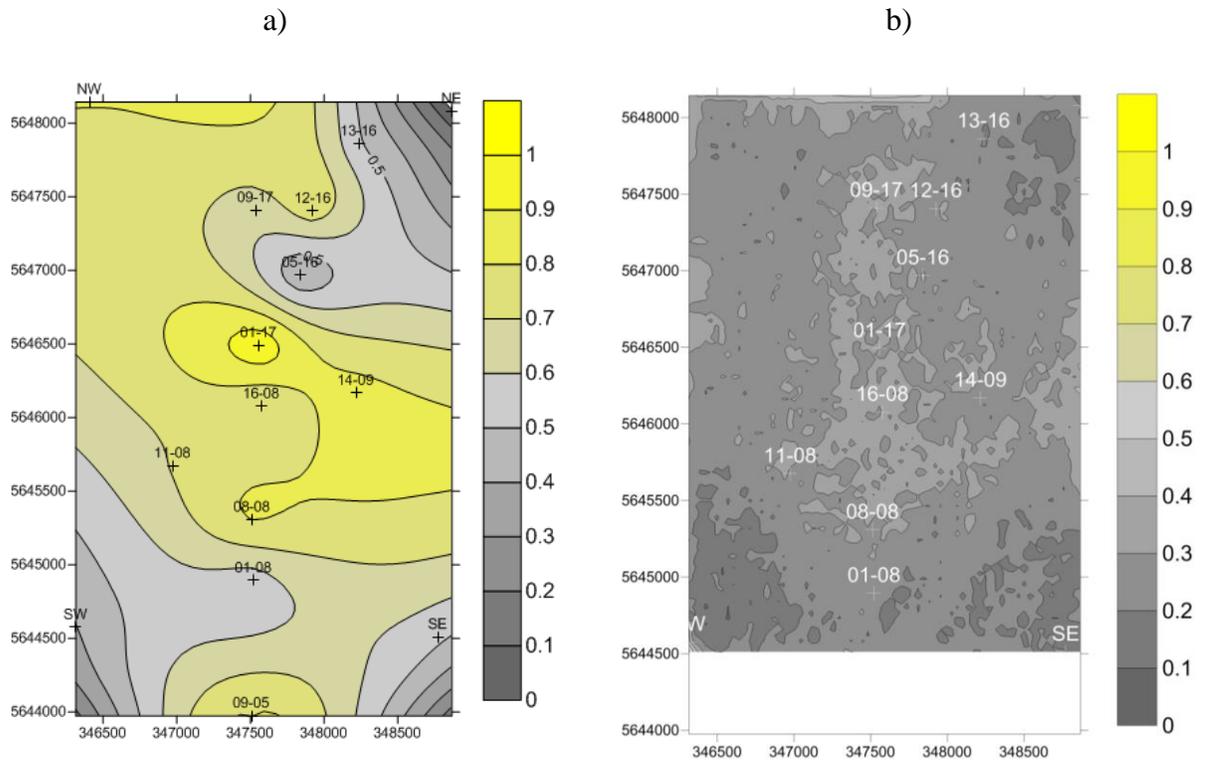


Figura 4.32: a) Mapa de probabilidad a priori (contenido de arena) derivado de la interpretación petrofísica b) Mapa de contenido de arena derivado de la clasificación por PDF bivariadas de atributos sísmicos correspondientes a la combinación Impedancia Acústica Relativa vs Primera Derivada.

## CONCLUSIONES Y RECOMENDACIONES

Hacerle un control de calidad a los pozos es sumamente importante, tomar detalles tan pequeños como la inclinación de los pozos o hacer una mejora de los registros permitieron que los pozos no tuviesen errores en su ubicación y que su interpretación petrofísica hubiese sido más sencilla de realizar.

Con la interpretación petrofísica, se pudieron identificar estratos con arenas que según (Yang, et al., 1996) pertenecen a cuerpos arenosos en la parte norte y a un canal de arena en la zona sur. Además El contenido de arena de la zona de estudio en el intervalo de interés fue relativamente alta entre el 50 y 99%, a excepción de los pozos ubicados en la parte noreste.

Los únicos registros petrofísicos que permitieron realizar una discriminación del contenido de arena fueron el Sónico P, Densidad Volumétrica e Impedancia Acústica, siendo este último el que mejor clasificaba. El comportamiento generalizado fue que las clasificaciones de arenas eran mejores en los pozos con alta intercalaciones de arenas y lutitas y no aquellos asociados a los cuerpos arenosos y al canal de arena.

Las curvas PDF bivariadas derivadas de la interpretación de los registros de pozos mostraron una clasificación excepcional de más del 80% en todos los pozos, en donde la probabilidad a priori tomó un papel muy importante, ya que llevó a estos valores de discriminación a ese nivel de acertación.

El contenido de arena y lutita o probabilidad a priori litológica, modificó drásticamente los porcentajes de clasificación de las PDF asociadas a la interpretación petrofísica.

Se pudo identificar la presencia de gas y agua en la mitad de los pozos, pero al generar las PDF sin tomar en cuenta las probabilidades a priori, el único fluido clasificado fue el agua, a excepción

de un pozo que solo presentaba gas. Luego de tomar en cuenta las probabilidades priori, en todos los pozos el único grupo clasificado correctamente fue el de los estratos lutíticos.

De los 11 atributos evaluados con las PDFs, solo tres atributos, Amplitud RMS, frecuencia dominante y Caos pudieron discriminar como máximo el 75% de los pozos. En cuanto a los gráficos bivariados derivados de la combinación de estos atributos, el porcentaje de clasificación fue mayor al 70%, destacando que la probabilidad a priori no generó ningún cambio sustancial en los porcentajes de clasificación

Para la clasificación litológica de atributos sísmicos, las PDF univariadas promedios no generaron ninguna clasificación litológica aceptable. Por otro lado, para los gráficos bivariados, de los 55 resultantes, solo 4 combinaciones presentaron buenos porcentajes de clasificación mayor al 80%, resaltando que la probabilidad litológica a priori no generó ningún cambio importante en los porcentajes de discriminación.

De las 4 combinaciones de atributos, solo dos presentaron tendencias en los valores de contenido de arena estimado, resaltando que este contenido de arena dista entre un 27 y 50% de los valores de contenido de arena estimados con interpretación petrofísica.

Existen muchas razones que podrían estar afectando los resultados esperados con el análisis de atributos. Una de ellas, puede estar asociado al hecho de que los atributos sísmicos no están respondiendo a cambios litológicos, como se pudo evidenciar en la PDFs univariadas asociadas a los atributos sísmicos (página 76, 4.7.1 - Curvas PDFs univariadas). Otro factor que puede estar afectando, es que para la clasificación litológica con los atributos sísmicos fueron usadas PDFs promedios que no necesariamente representan el comportamiento individual de cada pozo. Se recomienda generar una variedad más amplia de atributos sísmicos y si es conseguido algunos que puedan diferenciar litologías, hacer uso de algún método alternativo que no haga uso de las PDFs promedio, como por ejemplo el método de Similitud presentado por (Michelena, González, &

Mariangela , 1998) en donde se podría identificar zonas similares a ciertos pozos que permitan asignarles un porcentajes de probabilidad a priori más acorde.

Otra recomendación importante sería, hacer un estudio con una variedad más amplia de registros de pozos, con el fin de estudiar si existe algún otro registro o psuedo-registro que haga una buena discriminación litógica.

Por último, para reducir la incertidumbre se recomienda tener información de núcleos ya que estos presentan valores más confiables que los registros de pozos.

## BIBLIOGRAFÍA

- Avseth, P., Mukerji, T., & Mavko, G. (2010). *Quantitative Seismic Interpretation: Applying Rock Physics Tools to Reduce Interpretation Risk*. Cambridge University Press.
- Bautista, C. R. (2011). *Punto de Conversión común para la obtención de imágenes sísmicas de onda convertida (PS) a partir de sísmica multicomponente*. Bogotá.
- Chopra, S., & Marfurt, K. J. (s.f.). *Seismic Attributes for Prospect Identification and Reservoir Characterization* (SEG Geophysical Development Series ed., Vol. 11). 2007.
- Coxeter, H. S. (1989). *Introduction to Geometry* (2da ed.). Wiley.
- Cyders, T. (2012). *Notas del curso: Data Analysis*. Ohio University, Athens. EE.UU.
- Falcón, R. (2011). *Notas del curso de Tópicos en Estratigrafía Secuencial*. Universidad Simón Bolívar, Departamento de Ciencias de la Tierra, Caracas.
- Glover, P. (2014). *Petrophysics MSc Course Notes*. Department of Geology and Petroleum Geology, University of Aberdeen.
- Graps, A. (01 de Junio de 1995). An Introduction to Wavelets. *IEEE Computational Science & Engineering*, 2(2), 50 - 61.
- Hayes, B. J., Christopher, J. E., Rosenthal, L., Los, G., McKercher, B., Minken, D., . . . Fennell, J. (1994). Chapter 19: Cretaceous Mannville Group of the Western Canada Sedimentary Basin. En G. Mossop, & I. Shetsen, *Geological Atlas of the Western Canada Sedimentary Basin*. Canadian Society of Petroleum Geologists and the Alberta Research Council.
- Kay, K. (2014). *Clases del curso: Statistics and data analysis in Python*. Stanford University, California. EE.UU.
- Lacedre Ricciardi, G. J. (2014). *Inversión de datos multicomponente, campo blackfoot, Canadá*.
- Larsen, J. A. (1999). *AVO Inversion by Simultaneous P-P and P-S Inversion*. Department of Geology and Geophysics, University of Calgary.

- López Casuso, R. (2009). *Cálculo de Probabilidades e Inferencia Estadística con tópicos de Econometría* (Quinta ed.). Caracas: Publicaciones UCAB.
- Lu , H.-x., & Margrave, G. F. (1998). Reprocessing the Blackfoot 3C-3D seismic data. En *CREWES Research Report* (Vol. 10). Calgary, Canadá: Universidad de Calgary, Departamento de Geociencias.
- Massaad, J. (2014). *Análisis Avo 3d Basado En El Uso De Apilados Parciales En El Dominio De Distancia Fuente-Receptor*. Universidad Simón Bolívar, Coordinación de Ingeniería Geofísica, Caracas.
- Mendoza, J., & Landa, A. (2005). *Manual de PETROLUKE*. Universidad Simón Bolívar, Departamento de Ciencias de la Tierra, Caracas.
- Michelena, R. J., González, E. S., & Mariangela , C. (Abril de 1998). Similarity analysis: A new tool to summarize seismic attributes information. *The Leading Edge*.
- Mossop, G. D., & Shetsen, I. (1994). *The Geological Atlas of the Western Canada Sedimentary Basin*. Canada: Canadian Society of Petroleum Geologists and the Alberta Research Council.
- Pérez, F., & Granger, B. E. (2007). IPython: A System for Interactive Scientific Computing, Computing in Science and Engineering. *Computing in Science and Engineering*, 3, 21-29.
- Price, R. (1994). Chapter 2: Cordilleran Tectonics and the Evolution of the Western Canada Sedimentary Basin. En G. Mossop, & I. Shetsen, *Geological Atlas of the Western Canada Sedimentary Basin*.
- Regueiro, J. (2007). *Sísmica para Ingenieros y Geólogos: El Método Sísmico de Reflexión*.
- Ribeiro, M. I. (2004). *Gaussian Probability Density Functions: Properties and Error Characterization*. Instituto Superior Tecnico, Institute for Systems and Robotics, Lisboa. Portugal.
- Rider, M. (1996). *The Geological Interpretation of Well Logs* (Segunda ed.). Shutherland, Escocia: Whittles Publishing.

- Schlumberger Limited. (1991). *Log Interpretation Principles/Application*. Schlumberger Educational Services.
- Sheriff, R. E. (2002). *Encyclopedic Dictionary of Applied Geophysics* (4 ed.). Society of Exploration Geophysicists.
- Todorov, T. I. (2000). *Integration of 3C-3D seismic data and well logs for rock property estimation*. Department of Geology and Geophysics, University of Calgary.
- Yang, G., Lawton, D., Stewart, R., Miller, S., Potter, C., & Simin, V. (1996). Interpretation and analysis of the Blackfoot 3C-3D seismic survey. En *CREWES Research Report* (Vol. 8). Universidad de Calgary, Departamento de Geociencias.
- Yilmaz, O. (2008). *Seismic Data Analysis. Processing, Inversion and Interpretation of Seismic Data* (Vol. 1). Society of Exploration Geophysicists.

## APÉNDICE

**Apéndice A. Tablas asociadas a matrices de decisión bayesiana para la discriminación litológica de las PDF univariadas.**

Pozos	Sin la probabilidad a priori		Con la probabilidad a priori	
	Lutita clasificada como lutita	Arena clasificada como arena	Lutita clasificada como lutita	Arena clasificada como arena
12-16	0.71	0.85	0.58	0.93
09-17	0.66	0.66	0.43	0.83
01-08	0.77	0.54	0.93	0.28
01-17	0.69	0.55	0	1
05-16	0.66	0.74	0.95	0.27
08-08	0.52	0.63	0.18	0.92
09-05	0.7	0.61	0.49	0.77
11-08	0.8	0.53	0.84	0.48
13-16	0.63	0.7	0.97	0.2
14-09	0.65	0.61	0.43	0.79
16-08	0.78	0.41	0.87	0.3

Tabla 0.1: Matrices de decisión Bayesiana simplificada para el registro de sónico P.

Pozos	Sin la probabilidad a priori		Con la probabilidad a priori	
	Lutita clasificada como lutita	Arena clasificada como arena	Lutita clasificada como lutita	Arena clasificada como arena
12-16				
09-17	0.72	0.54	0.26	0.88
01-08	0.71	0.58	0.6	0.68
01-17	0.78	0.52	0	1
05-16	0.55	0.76	0.65	0.65

<b>08-08</b>			0	1
<b>09-05</b>	0.68	0.58	0	1
<b>11-08</b>	0.9	0.45		
<b>13-16</b>				
<b>14-09</b>	0.72	0.52	0	1
<b>16-08</b>	0.83	0.45	0.02	0.99

Tabla 0.2: Matrices de decisión Bayesiana simplificada para el registro de impedancia acústica.

	Sin la probabilidad a priori		Con la probabilidad a priori	
<b>Pozos</b>	<b>Lutita clasificada como lutita</b>	<b>Arena clasificada como arena</b>	<b>Lutita clasificada como lutita</b>	<b>Arena clasificada como arena</b>
<b>12-16</b>				
<b>09-17</b>				
<b>01-08</b>	0.69	0.52	1	0.02
<b>01-17</b>	0.83	0.49	0	1
<b>05-16</b>	0.72	0.49	0.98	0.17
<b>08-08</b>	0.47	0.68	0.22	0.9
<b>09-05</b>	0.34	0.79	0.15	0.93
<b>11-08</b>	0.71	0.42	0.83	0.29
<b>13-16</b>			0.98	0.08
<b>14-09</b>	0.76	0.43		
<b>16-08</b>	0.78	0.6	0.83	0.53

Tabla 0.3: Matrices de decisión Bayesiana simplificada para el registro de densidad bulk.

	Sin la probabilidad a priori		Con la probabilidad a priori	
<b>Pozos</b>	<b>Lutita clasificada como lutita</b>	<b>Arena clasificada como arena</b>	<b>Lutita clasificada como lutita</b>	<b>Arena clasificada como arena</b>
<b>12-16</b>				
<b>09-17</b>	0.74	0.67	0.3	0.97

<b>01-08</b>	0.99	0.98	0.99	0.98
<b>01-17</b>				
<b>05-16</b>	1	0.93	1	0.93
<b>08-08</b>				
<b>09-05</b>	0.89	0.77	0.39	0.99
<b>11-08</b>	1	0.98	0.98	1
<b>13-16</b>	0.96	0.81	0.94	0.83
<b>14-09</b>	0.96	0.75	0.38	1
<b>16-08</b>	0.87	0.77	0.41	0.98

Tabla 0.4: Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Sónico P e Impedancia acústica.

Pozos	Sin la probabilidad a priori		Con la probabilidad a priori	
	Lutita clasificada como lutita	Arena clasificada como arena	Lutita clasificada como lutita	Arena clasificada como arena
<b>12-16</b>				
<b>09-17</b>				
<b>01-08</b>	0.91	0.81	0.81	0.89
<b>01-17</b>	0.99	0.93	0.51	1
<b>05-16</b>	0.99	0.97	0.99	0.97
<b>08-08</b>	0.96	0.9	0.64	1
<b>09-05</b>	0.87	0.56	0.25	1
<b>11-08</b>	0.98	0.93	0.91	1
<b>13-16</b>	0.98	0.88	0.91	0.96
<b>14-09</b>	0.96	0.82	0.37	1
<b>16-08</b>	0.8	0.79	0.3	0.99

Tabla 0.5: Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Sónico P y Densidad Volumétrica.

Pozos	Sin la probabilidad a priori		Con la probabilidad a priori	
	Lutita clasificada como lutita	Arena clasificada como arena	Lutita clasificada como lutita	Arena clasificada como arena
<b>12-16</b>				
<b>09-17</b>	0.84	0.64	0.4	0.97
<b>01-08</b>	0.99	0.98	0.99	0.99
<b>01-17</b>	0.99	0.93	0.46	1
<b>05-16</b>	0.97	0.97	0.98	0.96
<b>08-08</b>	1	0.97	0.87	1
<b>09-05</b>	0.9	0.47	0.22	1
<b>11-08</b>	1	0.98	0.97	1
<b>13-16</b>	0.93	0.88	0.85	0.97
<b>14-09</b>	0.93	0.84	0.39	1
<b>16-08</b>	0.75	0.66	0.22	1

Tabla 0.6: Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Impedancia Acústica y Densidad Volumétrica.

**Apéndice B. Tablas asociadas a matrices de decisión bayesiana para la discriminación litológica de las PDFs bivariadas.**

Pozos	Arena clasificada como arena	
	Sin la probabilidad a priori	Con la probabilidad a priori
12-16	n/a	n/a
09-17	0.67	0.97
01-08	0.98	0.98
01-17	n/a	n/a
05-16	0.93	0.93
08-08	n/a	n/a
09-05	0.77	0.99
11-08	0.98	1.00
13-16	0.81	0.83
14-09	0.75	1.00
16-08	0.77	0.98

Tabla 0.7: Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Sónico P e Impedancia acústica.

Pozos	Arena clasificada como arena	
	Sin la probabilidad a priori	Con la probabilidad a priori
<b>12-16</b>	n/a	n/a
<b>09-17</b>	n/a	n/a
<b>01-08</b>	0.81	0.89
<b>01-17</b>	0.93	1.00
<b>05-16</b>	0.97	0.97
<b>08-08</b>	0.90	1.00
<b>09-05</b>	0.56	1.00
<b>11-08</b>	0.93	1.00
<b>13-16</b>	0.88	0.96
<b>14-09</b>	0.82	1.00
<b>16-08</b>	0.79	0.99

Tabla 0.8: Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Sónico P y Densidad Volumétrica.

Pozos	Arena clasificada como arena	
	Sin la probabilidad a priori	Con la probabilidad a priori
<b>12-16</b>	n/a	n/a
<b>09-17</b>	0.64	0.97
<b>01-08</b>	0.98	0.99
<b>01-17</b>	0.93	1.00
<b>05-16</b>	0.97	0.96
<b>08-08</b>	0.97	1.00
<b>09-05</b>	0.47	1.00
<b>11-08</b>	0.98	1.00
<b>13-16</b>	0.88	0.97
<b>14-09</b>	0.84	1.00
<b>16-08</b>	0.66	1.00

Tabla 0.9: Matrices de decisión Bayesiana simplificada para la representación bivariada de los registros Impedancia Acústica y Densidad Volumétrica.

## Apéndice C. Código para hallar las coordenadas de los pozos en la zona de interés

Archivo pozos\_vectores.txt.

```
1. Well  x_top  y_top  tvd_top x_bottom  y_bottom  tvd_bottom  x_zone  y_zone  tvd_zone
2. 12-
   16  347733.50  5647580.90  143368  347944.00  5647387.80  1661806  347925.7278262267
   5647404.5617898125  -1530.0
3. 09-
   17  347733.70  5647587.90  196174  347517.40  5647388.10  1676985  347538.8698941999
   5647407.932107541  -1530.0
4. 01-
   08  347650.00  5644633.40  136698  347512.10  5644916.70  1653850  347523.35722076625
   5644893.573309332  -1530.0
5. 01-
   17  347708.20  5646195.60  152143  347543.90  5646522.70  1675758  347559.61790734535
   5646491.407684159  -1530.0
6. 05-
   16  347733.30  5647573.90  151814  347852.60  5646909.80  1661985  347842.1734916774
   5646967.840605004  -1530.0
7. 08-
   08  347398.40  5645690.30  145800  347525.90  5645264.60  1685290  347513.0389388694
   5645307.540813516  -1530.0
8. 09-
   05  347674.00  5644632.70  201748  347496.40  5643927.90  1624075  347508.1467502199
   5643974.5166078545  -1530.0
9. 11-
   08  347384.30  5645690.80  190370  346930.40  5645671.70  1678185  346975.6080208225
   5645673.602342362  -1530.0
10. 13-
   16  348368.20  5648288.40  167365  348222.70  5647809.20  1693925  348238.324074717
   5647860.657433707  -1530.0
11. 14-
   09  348032.40  5646152.00  210591  348231.60  5646169.80  1632359  348217.2587623297
   5646168.518503862  -1530.0
12. 16-
   08  347720.20  5646195.20  141501  347565.70  5646066.90  1636177  347576.6751855921
   5646076.014021434  -1530.0
```

Archivo main.py.

```
1. # -*- coding: utf-8 -*-
2. from __future__ import absolute_import, print_function, division, unicode_literals
3.
4. import pandas as pd
5. from planeLine import LinePlane
```

```

6.
7. # Plano de intersección
8. p_co = [0, 0, -1530.000]
9. p_no = [0, 0, 1]
10.
11. vector = pd.read_csv("pozos_vectores.txt", header=0, delim_whitespace=True)
12.
13. for index, row in vector.iterrows():
14.     # Puntos extrmos del vector/pozo
15.     p0 = [row['x_top'], row['y_top'], -row['tvd_top']]
16.     p1 = [row['x_bottom'], row['y_bottom'], -row['tvd_bottom']]
17.
18.     linePlane = LinePlane(p0, p1, p_co, p_no)
19.
20.     print(linePlane.intersection())

```

### Archivo planeLine.py

```

1. from __future__ import print_function, division, unicode_literals
2.
3. '''
4. Extraido y adaptado de:
5. - http://stackoverflow.com/questions/5666222/3d-line-plane-intersection#answer-18543221
6. '''
7.
8. class LinePlane:
9.     """
10.    p0, p1: define the line
11.    p_co, p_no: define the plane:
12.        p_co is a point on the plane (plane coordinate).
13.        p_no is a normal vector defining the plane direction; does
14.            not need to be normalized.
15.    """
16.
17.    def __init__(self, p0, p1, p_co, p_no):
18.        self.p0 = p0
19.        self.p1 = p1
20.        self.p_co = p_co
21.        self.p_no = p_no
22.        self.epsilon = 0.000001
23.
24.    # generic math functions
25.
26.    def add(self, a, b):
27.        return [a[0] + b[0],
28.                a[1] + b[1],
29.                a[2] + b[2]]
30.
31.    def subtract(self, a, b):
32.        return [a[0] - b[0],
33.                a[1] - b[1],
34.                a[2] - b[2]]
35.
36.    def dotProduct(self, a, b):
37.        return (a[0] * b[0] +
38.                a[1] * b[1] +
39.                a[2] * b[2])
40.
41.    def multiply(self, a, f):
42.        return [a[0] * f,
43.                a[1] * f,

```

```

44.         a[2] * f]
45.
46.     def intersection(self):
47.         """
48.         Intersect line/plane.
49.         return a Vector or None (when the intersection can't be found).
50.         """
51.
52.         u = self.subtract(self.p1, self.p0)
53.         w = self.subtract(self.p0, self.p_co)
54.         dot = self.dotProduct(self.p_no, u)
55.
56.         if abs(dot) > self.epsilon:
57.             # the factor of the point between p0 -> p1 (0 - 1)
58.             # if 'fac' is between (0 - 1) the point intersects with the segment.
59.             # otherwise:
60.             # < 0.0: behind p0.
61.             # > 1.0: infront of p1.
62.             fac = -self.dotProduct(self.p_no, w) / dot
63.             u = self.multiply(u, fac)
64.             return self.add(self.p0, u)
65.         else:
66.             # The segment is parallel to plane
67.             return None

```

## Apéndice D. Código para realizar la descomposición espectral

```
1. %*****
2. % Carga de datos Segy [Data, Segy,TraceHeader,SegyHeader]
3. % Estimacion de CWT ZambranoSV
4. %
5. %*****
6.
7. k=8:4:32; % Frecuencias
8. place=[25 16 12 10 8 7 6]; % Ubicacion dentro del vector de pseudo frecuencias
9.
10. s1='./cubos/VERTMIG';
11. s3='.segy';
12. s4='mono';
13. s5='VERTMIG_morlet';
14. s6='_waveTeke';
15. s7='./cubos/';
16.
17. import=strcat(s1,s3); % Lectura
18. [Data,SegyTraceHeader,SegyHeader]=ReadSegy(import);
19. cdp=[SegyTraceHeader.cdp];
20. cdp_trace=[SegyTraceHeader.cdpTrace];
21. dt=[SegyHeader.dt];
22. [ns,Ntrace]=size(Data); % Dimensiones del cubo
23.
24. for n =1:length(k);
25.     str=num2str(k(n));
26.     s2=str;
27.
28.     scales= 1:64;
29.     %     freq=scal2frq(scales,'mor1',dt/1E6);
30.     %     time=500:dt/1E3:2500;
31.     %     ft=s2; % Freq
32.     %     place=find(freq >=ft,1,'last');%Ubicacion de la freq en funcion de las escalas
33.     CWT_monofreq=zeros(ns,Ntrace); % Matriz Monofreq ceros
34.
35.     for i=1:Ntrace; % Ntrace
36.         coef=cwt(Data(:,i),scales,'mor1');
37.         CWT_monofreq(:,i)=coef(place(n,:),:);
38.     end
39.
40.     %Productos : Cubo Monofrecuencia
41.     exportM=strcat(s7,s5,s2,'Hz',s3);
42.
43.     WriteSegyStructure(exportM,SegyHeader,SegyTraceHeader,CWT_monofreq);
44.
45.     clear CWT_monofreq ft
46.     display(strcat('Ready ', num2str(n), ' from ', num2str(length(k)), ': ', num2str(k(
47. n)), ' frequency'))
47. end
```

## Apéndice E. Códigos para el manejo de registros de pozos

Archivo `__init__.py`:

```
1. from __future__ import print_function, division
2. import matplotlib as mpl
3.
4. mpl.rcParams['figure.edgecolor'] = 'grey'
5. mpl.rcParams['figure.facecolor'] = 'white'
6. mpl.rcParams['legend.loc'] = 'best'
7. # try:
8. #     mpl.rcParams['figure.figsize'] = '12, 6'
9. # except AssertionError:
10. #     pass
11. mpl.rcParams['savefig.dpi'] = 2 * mpl.rcParams['savefig.dpi']
12.
13. mpl.rcParams['axes.facecolor'] = '#eeeeee'
14. mpl.rcParams['axes.edgecolor'] = '#bcbcbc'
15. mpl.rcParams['axes.linewidth'] = '0.5'
16. mpl.rcParams['axes.labelcolor'] = '#555555'
17. mpl.rcParams['axes.grid'] = True
18. mpl.rcParams['grid.color'] = '0.95'
19. mpl.rcParams['grid.linestyle'] = '-'
20. mpl.rcParams['grid.linewidth'] = '2'
21. mpl.rcParams['ytick.color'] = '#555555'
22. mpl.rcParams['xtick.color'] = '#555555'
23. mpl.rcParams['axes.axisbelow'] = True
24. mpl.rcParams[
25.     'axes.color_cycle'] = '348ABD, 7A68A6, A60628, 467821, CF4457, 188487, E24A33'
26. mpl.rcParams['legend.fancybox'] = True
27.
28. from .las_reader import LASReader, LASItem
29. from .las_plot import LASPlot, LASPlotCustom, PlotPair, PlotZones
30. from .las_plot import ScatterHist, ContourBI, PlotPosority
31. from .las_tools import LASSelect, AddCurve, FilterData, RollingWindow, Despike, SmothData
32. from .las_tools import SetLitZones, SplitByLit, getValuesZone, AreaCurve
33. from .bayesian import PlotHistPDF, CreateMatrix
```

Archivo `bayesian.py`

```
1.
2.     #!/usr/bin/env python
3.     # -*- coding: utf-8 -*-
4.     """ Lorem ipsum dolor sit amet. """
5.
6.
7.     import math
8.     import numpy as np
9.     import matplotlib.pyplot as plt
10.    import scipy.stats as stats
11.
12.    from scipy import interpolate
13.    #from scipy import *
14.    #from scipy.interpolate import *
15.    import pandas as pd
```

```

16.
17.
18. bins = 30
19.
20.
21.
22. def PlotHistPDF(data, xlabel="X Label", title="Title", data_labels=None, prob=None):
23.     ''' Genera un histograma y una pdf de un set de datos
24.     '''
25.     global bins
26.
27.     Pdf_priori = []
28.     Pdf_nopriori = []
29.
30.     if data_labels is None:
31.         data_labels = ["Shale", "Sand-Shale",
32.                        "Sand-Clean", "Sand Gas", "Sand Water"]
33.
34.     prob_index = ['shale', 'sand_shale',
35.                  'sand_clean', 'sand_gas', 'sand_water']
36.
37.     # print(len(data_labels))
38.
39.     data_colors = ["gray", "orange", "yellow", "green", "blue"]
40.
41.     # bins = 30
42.     alpha = 0.5
43.
44.     fig, ax = plt.subplots(1, 3, figsize=(15, 5))
45.     # fig2, ax2 = plt.subplots(1, 2, figsize=(10, 5))
46.     # print("###-- INFO FOR", title.upper(), "VALUES: --###")
47.
48.     # print(len(data))
49.
50.     arr = range(len(data))
51.     arr_priori = range(len(data))
52.     for j in range(len(data)):
53.         xn = data[j]
54.         if len(xn) > 1:
55.             # Now, we create an instance of the gaussian_kde class and feed
56.             # our sample to it:
57.             gkde = stats.gaussian_kde(xn)
58.             # We need some points at which we evaluate the density
59.             # function for the estimated density function
60.             shift = (np.nanmax(xn) + abs(np.nanmin(xn))) * 0.03
61.             ind_min = math.floor(np.nanmin(xn) - shift)
62.             ind_max = math.ceil(np.nanmax(xn) + shift)
63.             ind_size = abs(abs(ind_min) - abs(ind_max)) + 1
64.
65.             if (ind_size < 1000) or (ind_size > 100000):
66.                 ind_size = 10000
67.
68.             ind = np.linspace(ind_min, ind_max, ind_size)
69.
70.             kdepdf = gkde.evaluate(ind)
71.
72.             # arr[j] = np.zeros((size(ind),2))
73.             arr[j] = np.column_stack((ind, kdepdf))
74.             arr_priori[j] = np.column_stack(
75.                 (ind, kdepdf * prob[prob_index[j]]))
76.
77.             # plot pdf no a-priori
78.             ax[1].plot(ind, kdepdf, color=data_colors[j],

```

```

79.             label=data_labels[j])
80.
81.         # plot pdf with a-priori
82.         ax[2].plot(
83.             ind, kdepdf * prob[prob_index[j]], color=data_colors[j],
84.             label=data_labels[j])
85.
86.
87.         Pdf_nopriori.append(np.array((ind, kdepdf)))
88.         Pdf_priori.append(np.array((ind, kdepdf * prob[prob_index[j]])))
89.
90.
91.         # plot histogram of sample
92.         ax[0].hist(xn, bins,
93.                   normed=1, facecolor=data_colors[j],
94.                   alpha=alpha * 1.2,
95.                   label=data_labels[j])
96.
97.     for i in [0, 0]:
98.         ax[i].legend(framealpha=0.5)
99.         ax[i].set_xlabel(xlabel)
100.        ax[i].set_ylabel('Frequency')
101.
102.    ax[1].set_title('Sin P-Apriori')
103.    ax[2].set_title('Con P-Apriori')
104.
105.        # ax2[i].legend(framealpha=0.5)
106.        # ax2[i].set_xlabel(xlabel)
107.        # ax2[i].set_ylabel('Frequency')
108.    #print (len(xPdf_nopriori))
109.
110.    #print(a[0, :])
111.
112.    return fig, ax, Pdf_nopriori, Pdf_priori
113.
114.
115. def get_extrapolated(data1, min_value, max_value):
116.     """
117.     min_value = min_value - 0.001
118.     max_value = max_value + 0.001
119.     num = 1000
120.
121.     pre_x = np.linspace(min_value, data1[0,0], num)
122.     pre_y = np.empty(num)
123.     pre_y[:] = 0.0
124.     pre_data = np.array([pre_x, pre_y]).T
125.
126.     interm_data = np.concatenate((pre_data, data1), axis=0)
127.
128.     post_x = np.linspace(data1[-1,0], max_value, num)
129.     post_y = np.empty(num)
130.     post_y[:] = 0.0
131.     post_data = np.array([post_x, post_y]).T
132.
133.     post_data = np.concatenate((interm_data, post_data), axis=0)
134.
135.     f = interpolate.interp1d(post_data[:,0], post_data[:,1],fill_value=0.0)
136.
137.     xx = np.linspace(post_data[0,0], post_data[-1,0], num)
138.     yy = f(xx)
139.
140.     return np.array((xx, yy)).T
141.

```

```

142.
143. def get_new_func(yy, yy2):
144.     .....
145.     new_yy = yy - yy2
146.     new_yy[new_yy > 0] = 1
147.     new_yy[new_yy <= 0] = 0
148.
149.     new_yy2 = yy2 - yy
150.     new_yy2[new_yy2 > 0] = 1
151.     new_yy2[new_yy2 <= 0] = 0
152.
153.     return new_yy, new_yy2
154.
155. def get_area(x2,y2):
156.     area = 0;
157.     for i in range(x2.size - 1):
158.         height = y2[i];
159.         base = x2[i+1] - x2[i] ;
160.         area = area + (base*height);
161.
162.     return area
163.
164.
165.
166. def get_matrix(data1, data2, data13, data23, curvesNames):
167.     .....
168.     if ((data1.shape[0] == data1.shape[1]) and (data2.shape[0] == data2.shape[1])):
169.         total_area_c1 = np.sum(data1)
170.         true_area_c1 = np.sum(data13)
171.
172.         total_area_c2 = np.sum(data2)
173.         true_area_c2 = np.sum(data23)
174.     else:
175.         total_area_c1 = get_area(data1[:,0],data1[:,1])
176.         true_area_c1 = get_area(data13[:,0],data13[:,1])
177.
178.         total_area_c2 = get_area(data2[:,0],data2[:,1])
179.         true_area_c2 = get_area(data23[:,0],data23[:,1])
180.
181.
182.     matrix = np.zeros((2,3))
183.     matrix[0,0] = (true_area_c1) / total_area_c1
184.     matrix[0,1] = (total_area_c1 - true_area_c1) / total_area_c1
185.     matrix[0,2] = (total_area_c1) / total_area_c1
186.
187.     matrix[1,0] = (total_area_c2 - true_area_c2) / total_area_c2
188.     matrix[1,1] = true_area_c2 / total_area_c2
189.     matrix[1,2] = total_area_c2 / total_area_c2
190.
191.     #matrix[2,0] = matrix[0,0] + matrix[1,0]
192.     #matrix[2,1] = matrix[0,1] + matrix[1,1]
193.     #matrix[2,2] = np.nan
194.
195.     matrix = np.round(matrix, 2)
196.
197.     df = pd.DataFrame(matrix, index=[curvesNames[0], curvesNames[1]], columns=[curvesN
ames[0], curvesNames[1], 'Total'])
198.     df = df.astype(object).replace(np.nan, '')
199.
200.     return matrix, df
201.
202.
203. #def get_matrixBI(data1, data2, data13, data23, curvesNames):

```

```

204. #     '.....'
205. #     total_area_c1 = np.sum(data1)
206. #     true_area_c1 = np.sum(data13)
207. #
208. #     total_area_c2 = np.sum(data2)
209. #     true_area_c2 = np.sum(data23)
210. #
211. #     matrix = np.zeros((2,3))
212. #     matrix[0,0] = (true_area_c1) / total_area_c1
213. #     matrix[0,1] = (total_area_c1 - true_area_c1) / total_area_c1
214. #     matrix[0,2] = (total_area_c1) / total_area_c1
215. #
216. #     matrix[1,0] = (total_area_c2 - true_area_c2) / total_area_c2
217. #     matrix[1,1] = true_area_c2 / total_area_c2
218. #     matrix[1,2] = total_area_c2 / total_area_c2
219. #
220. #     #matrix[2,0] = matrix[0,0] + matrix[1,0]
221. #     #matrix[2,1] = matrix[0,1] + matrix[1,1]
222. #     #matrix[2,2] = np.nan
223. #
224. #     matrix = np.round(matrix, 2)
225. #
226. #     df = pd.DataFrame(matrix, index=[curvesNames[0], curvesNames[1]], columns=[curves
Names[0], curvesNames[1], 'Total'])
227. #     df = df.astype(object).replace(np.nan, '')
228. #
229. #     return matrix, df
230. #
231.
232. def CreateMatrix(inputData, curvesNames):
233.     '.....'
234.
235.     data1 = inputData[0].T
236.     data2 = inputData[1].T
237.
238.     #data1[:,0].shape
239.
240.     #print(data1[:,0].shape)
241.     #print(data2[:,0].shape)
242.     #print("-----")
243.     #return
244.
245.     min_x = np.concatenate((data1[:,0], data2[:,0])).min()
246.     max_x = np.concatenate((data1[:,0], data2[:,0])).max()
247.
248.
249.     data1_extrapolated = get_extrapolated(data1, min_x, max_x)
250.     data2_extrapolated = get_extrapolated(data2, min_x, max_x)
251.
252.     #plot_data(data1, data2, data1_extrapolated, data2_extrapolated)
253.
254.
255.     [new_data12y, new_data22y] = get_new_func(data1_extrapolated[:,1], data2_extrapolated[:,1])
256.
257.     data1_trueValues = np.array((data1_extrapolated[:,0], data1_extrapolated[:,1]*new_data12y)).T
258.     data2_trueValues = np.array((data2_extrapolated[:,0], data2_extrapolated[:,1]*new_data22y)).T
259.
260.
261.     #plot_data(data1, data2, data1_trueValues, data2_trueValues)
262.

```

```

263.     [matrix,df] = get_matrix(data1, data2, data1_trueValues, data2_trueValues, curvesN
      ames)
264.
265.
266.     return matrix, df
267.
268.
269. def CreateMatrixBI(inputData, curvesNames):
270.     '''
271.     data1 = inputData[0]
272.     data2 = inputData[1]
273.
274.     #plot
275.
276.     [new_data12y, new_data22y] = get_new_func(data1, data2)
277.
278.     data1_trueValues = data1*new_data12y
279.     data2_trueValues = data2*new_data22y
280.
281.     #plot
282.
283.     [matrix,df] = get_matrix(data1, data2, data1_trueValues, data2_trueValues, curvesN
      ames)
284.
285.
286.     return matrix, df
287.
288.
289. def plot_data(data1, data2, data12, data22):
290.     #fig, ax = plt.subplots()
291.     fig, (ax1, ax2) = plt.subplots(1, 2, sharey=True)
292.
293.     ax1.plot(data1[:,0], data1[:,1], label = "curve 1", color = "gray")
294.     ax1.plot(data2[:,0], data2[:,1], label = "curve 2", color = "yellow")
295.
296.     ax1.set_xlabel("Property")
297.     ax1.set_ylabel('Frequency')
298.     ax1.legend()
299.     ax1.set_title('no interpolation')
300.     #ax1.set_ylim(-0.005, 0.025)
301.     #ax1.set_xlim(140, 450)
302.
303.     ax2.plot(data12[:,0], data12[:,1], label = "curve 1.1", color = "gray")
304.     ax2.plot(data22[:,0], data22[:,1], label = "curve 2.1", color = "yellow")
305.     ax2.set_xlabel("Property")
306.     ax2.set_ylabel('Frequency')
307.     ax2.legend()
308.     ax2.set_title('extrapolated')
309.     #ax2.set_ylim(-0.005, 0.025)
310.     #ax2.set_xlim(140, 450)
311.
312.     plt.show()
313.

```

Archivo las\_plot.py

```

1. #!/usr/bin/env python
2. # -*- coding: utf-8 -*-
3.
4. """ Lorem ipsum dolor sit amet."""

```

```

5.
6. import numpy as np
7.
8. import matplotlib.pyplot as plt
9. from matplotlib.ticker import NullFormatter
10.
11. from .bayesian import PlotHistPDF, CreateMatrix, CreateMatrixBI
12.
13. from .las_tools import SplitByLit
14. bins = 30
15. ticks_labels = {}
16. ticks_labels["GR"] = [0, 50, 100, 150, 200] # Units: API
17. ticks_labels["GRS"] = ticks_labels["GR"]
18. # ticks_labels["RHOB"] = [2000, 3000] # Units: KG/M3
19. ticks_labels["NPSS"] = [0, 0.4, 0.8, 1.0] # Units: FRACTION
20. ticks_labels["PHIN"] = ticks_labels["NPSS"] # Units: FRACTION
21. ticks_labels["PHINS"] = ticks_labels["PHIN"] # Units: FRACTION
22. ticks_labels["PHIDS"] = ticks_labels["PHIN"]
23. # ticks_labels["PHIA"] = ticks_labels["NPSS"] # Units: FRACTION
24. # ticks_labels["PHID"] =
25. # ticks_labels["NPSS"] # Units:
26. # FRACTION
27. ticks_labels["DPSS"] = ticks_labels["NPSS"] # Units: FRACTION
28. ticks_labels["ILD"] = [0, 10, 100, 1000] # Units: OHMM
29. ticks_labels["ILDS"] = ticks_labels["ILD"] # Units: OHMM
30. ticks_labels["VSH"] = [0, 0.4, 1] # Units: OHMM
31. # ticks_labels["DT"] = [100, 300, 500, 700, 900] # Units: us/m
32. ticks_labels["NPHI"] = ticks_labels["NPSS"] # Units: v/v
33. ticks_labels["RHOB2"] = [1, 2, 3] # Units: gr/cc
34. ticks_labels["RHOB"] = [1000, 2000, 3000] # Units: kg/m3
35. ticks_labels["RHOB3"] = ticks_labels["RHOB"] # Units: kg/m3
36. color_curve = {}
37. color_curve["DT"] = "black"
38. color_curve["DTS"] = color_curve["DT"]
39. color_curve["GR"] = "red"
40. color_curve["GRS"] = color_curve["GR"]
41. color_curve["RHOB"] = "red"
42. color_curve["RHOB3"] = color_curve["RHOB"]
43. color_curve["VSH"] = "brown"
44. color_curve["PHID"] = "green"
45. color_curve["PHIDS"] = color_curve["PHID"]
46. color_curve["PHIN"] = color_curve["ILD"] = "blue"
47. color_curve["PHINS"] = color_curve["PHIN"]
48.
49. log_logs = ["ILD", "ILDS"]
50. por_logs = ["NPSS", "PHIN", "PHIA", "PHID", "PHINS", "PHIDS"]
51.
52. fontProperties = {'family': 'sans-serif', 'sans-serif': ['Helvetica'],
53.                  'weight': 'normal', 'size': 10}
54.
55.
56. def plot_tops(x, ax, tops):
57.     """ Plot Top and Bottom Lines """
58.     for top in tops:
59.         plt.hlines(top["value"],
60.                   np.nanmin(x),
61.                   np.nanmax(x),
62.                   label=top["label"])
63.         ax.text(np.nanmin(x), top["value"] - 2, top[
64.                 "label"], fontsize=6, color="black")
65.
66.
67. def LASPlot(well, logs=None, **kwargs):

```

```

68. # TODO: Check if len(logs) is less than len(logames)
69. """Plot well logs """
70. if (logs is None) or (type(logs) is list):
71.     logNames = well.curves.names
72.     totalLogs = len(logNames)
73.     totalPlots = len(logs)
74.     log = 1
75.     name = 0
76.     fig = axes = False
77.     pos = {}
78.
79.     # TODO: keep order of 'logs' list
80.     for name in range(totalLogs):
81.         logName = logNames[name]
82.         curves = well.curves.deliver()
83.
84.         if not (logName in logs):
85.             continue
86.
87.         if (logName != 'DEPTH'):
88.             xy = np.vstack((well.data['DEPTH'], well.data[logName])).T
89.
90.             if "zone" in kwargs:
91.                 # TODO: Check if kwargs['zone'][0] is lower than
92.                 # kwargs['zone'][1]
93.                 cond = np.logical_and(xy[:, 0] >= kwargs['zone'][0],
94.                                       xy[:, 0] <= kwargs['zone'][1])
95.                 xy = xy[cond]
96.
97.             if "sharey" in kwargs:
98.                 sharey = True
99.             else:
100.                 sharey = False
101.
102.             if (log == 1):
103.                 if (totalPlots == 1):
104.                     fig = plt.figure()
105.                     axes = fig.add_axes([0.1, 0.1, 0.5, 1])
106.                 else:
107.                     fig, axes = plt.subplots(
108.                         1, totalPlots, figsize=(10, 8), sharey=sharey)
109.
110.             if (totalPlots == 1):
111.                 ax = axes
112.             else:
113.                 try:
114.                     ax = axes[log - 1]
115.                 except:
116.                     break
117.
118.             ax.invert_yaxis()
119.
120.             if (logName in log_logs):
121.                 ax.set_xscale('log')
122.
123.             ax.set_xlabel(
124.                 logName + " (" + curves[logName]['units'].lower() + ")")
125.             ax.set_ylabel(
126.                 "Depth (" + curves["DEPTH"]['units'].lower() + ")")
127.             ax.set_ylim(kwargs['zone'][1] + 10, kwargs['zone'][0] - 10)
128.             ax.set_title(logName + " vs Depth")
129.
130. # Plot

```

```

131.         if logName in color_curve:
132.             curve_color = color_curve[logName]
133.         else:
134.             curve_color = "black"
135.
136.         # if logName in por_logs:
137.         # ax.set_xlim(0, 1)
138.
139.         ax.plot(xy[:, 1],
140.                xy[:, 0],
141.                curve_color,
142.                lw=0.5, label=logName)
143.
144.         pos[log] = logName
145.         log += 1
146.
147.         if "returnpos" in kwargs:
148.             if kwargs["returnpos"] is True:
149.                 return fig, axes, pos
150.             else:
151.                 return fig, axes
152.
153.         else:
154.             print "Warning: variable log must be a list"
155.
156.
157. def LASPlotCustom(x, y, xlabel, ylabel, title, color, tops=[], xinverse=False):
158.     ''' Genera un gráfico especial para un registro de pozo
159.     '''
160.
161.     fig = plt.figure()
162.     ax = fig.add_axes([0.1, 0.1, 0.5, 1])
163.     # left, bottom, width, height (range 0 to 1)
164.     # ax.grid()
165.     ax.set_xlabel(xlabel)
166.     ax.set_ylabel(ylabel)
167.     ax.set_title(title)
168.     ax.invert_yaxis()
169.
170.     if xinverse:
171.         ax.invert_xaxis()
172.     ax.plot(x, y, c=color, lw=0.5)
173.
174.     return fig, ax
175.
176.
177. def PlotPair(well, data_input, lit=None, prob=None, createMatrix =False):
178.     '''
179.     Lorem
180.     '''
181.
182.     matrix_aprioir = []
183.     matrix_no_apriori = []
184.
185.     curves = well.curves.deliver()
186.     if lit is None:
187.         lit = well.data["LIT"]
188.
189.     var = [0, 0]
190.     var[0] = well.data[data_input[0]]
191.
192.     label = [0, 0]
193.

```

```

194.     label[0] = data_input[0] + \
195.         " (" + curves[data_input[0]]['units'].lower() + ")"
196.
197.     if len(data_input) == 2:
198.         var[1] = well.data[data_input[1]]
199.         RHOB_VP = SplitByLit(var[0], var[1], lit)
200.
201.         fig = plt.figure()
202.         ax = fig.add_subplot(111)
203.
204.         ax.scatter(RHOB_VP[0][:, 1], RHOB_VP[0][
205.             :, 0], c='gray', s=8, lw=0.1, label="Shale")
206.         ax.scatter(RHOB_VP[1][:, 1], RHOB_VP[1][:, 0],
207.             c='orange', s=8, lw=0.1, label="Sand-Shale")
208.         ax.scatter(RHOB_VP[2][:, 1], RHOB_VP[2][
209.             :, 0], c='yellow', s=8, lw=0.1, label="Sand-Clean")
210.         ax.scatter(RHOB_VP[3][:, 1], RHOB_VP[3][
211.             :, 0], c='green', s=8, lw=0.1, label="Gas")
212.         ax.scatter(RHOB_VP[4][:, 1], RHOB_VP[4][
213.             :, 0], c='blue', s=8, lw=0.1, label="Water")
214.
215.         plt.title(data_input[1] + ' vs ' + data_input[0])
216.
217.         label[1] = data_input[1] + \
218.             " (" + curves[data_input[1]]['units'].lower() + ")"
219.
220.         plt.xlabel(label[1])
221.         plt.ylabel(label[0])
222.
223.         plt.legend(framealpha=0.5)
224.
225.     else:
226.         RHOB_VP = SplitByLit(var[0], var[0], lit)
227.
228.         for i in range(len(data_input)):
229.             d_sh = RHOB_VP[0][:, i]
230.             d_sns = RHOB_VP[1][:, i]
231.             d_sn_cl = RHOB_VP[2][:, i]
232.             d_sn_gs = RHOB_VP[3][:, i]
233.             d_sn_wt = RHOB_VP[4][:, i]
234.             vp_data = [d_sh, d_sns, d_sn_cl, d_sn_gs, d_sn_wt]
235.
236.             [fig, ax, Pdf_nopriori, Pdf_priori] = PlotHistPDF(vp_data, xlabel=label[i], ti
237.                 tle=data_input[i], prob=prob)
238.
239.             if (createMatrix):
240.                 #print("No a-priori")
241.                 #data1 = Pdf_nopriori[:,0].T
242.                 #data2 = Pdf_nopriori[:,1].T
243.
244.                 #print(data1[:,0])
245.                 #print(data2[:,0])
246.
247.                 [matrix,df] = CreateMatrix(Pdf_nopriori, ["Shale", "Sand"])
248.                 #print(df)
249.                 #df.plot(title='No A-priori')
250.                 matrix_no_apriori.append(df)
251.
252.                 #print("A-priori")
253.                 [matrix,df] = CreateMatrix(Pdf_priori, ["Shale", "Sand"])
254.                 #print(df)
255.                 #df.plot(title='A-priori')
256.                 matrix_apriori.append(df)

```

```

256.
257.
258.     plt.show()
259.     return matrix_no_apriori, matrix_apriori
260.
261. def PlotPosority(well, data_input, lit=None):
262.     '''
263.     Lorem
264.     '''
265.     curves = well.curves.deliver()
266.     if lit is None:
267.         lit = well.data["LIT"]
268.
269.     var = [0, 0]
270.     var[0] = well.data[data_input[0]]
271.
272.     label = [0, 0]
273.
274.     label[0] = data_input[0] + \
275.         " (" + curves[data_input[0]]['units'].lower() + ")"
276.
277.     if len(data_input) == 2:
278.         var[1] = well.data[data_input[1]]
279.         RHOB_VP = SplitByLit(var[0], var[1], lit)
280.
281.         fig = plt.figure()
282.         ax = fig.add_subplot(111)
283.
284.         ax.scatter(RHOB_VP[0][:, 1], RHOB_VP[0][:, 0], c='red', s=5, lw=0,
285.                 label=r'$0.33 < \phi <= 1$')
286.         ax.scatter(RHOB_VP[1][:, 1], RHOB_VP[1][:, 0],
287.                 c='orange', s=5, lw=0,
288.                 label=r'$0.24 < \phi <= 0.33$', marker='v')
289.         ax.scatter(RHOB_VP[2][:, 1], RHOB_VP[2][:, 0], c='blue', s=5, lw=0,
290.                 label=r'$0.16 < \phi <= 0.24$', marker='s')
291.         ax.scatter(RHOB_VP[3][:, 1], RHOB_VP[3][:, 0], c='green', s=5, lw=0,
292.                 label=r'$0.08 < \phi <= 0.16$', marker='s')
293.         ax.scatter(RHOB_VP[4][:, 1], RHOB_VP[4][:, 0], c='gray', s=5, lw=0,
294.                 label=r'$0.00 < \phi <= 0.08$', marker='s')
295.
296.         plt.title(data_input[1] + ' vs ' + data_input[0])
297.
298.         label[1] = data_input[1] + \
299.             " (" + curves[data_input[1]]['units'].lower() + ")"
300.
301.         plt.xlabel(label[1])
302.         plt.ylabel(label[0])
303.
304.         plt.legend(framealpha=0.4, fontsize=8)
305.
306.     else:
307.         RHOB_VP = SplitByLit(var[0], var[0], lit)
308.
309.     v_data_sets_labels = [r'$0.33 < \phi <= 1$',
310.                          r'$0.24 < \phi <= 0.33$',
311.                          r'$0.16 < \phi <= 0.24$',
312.                          r'$0.08 < \phi <= 0.16$',
313.                          r'$0.00 < \phi <= 0.08$']
314.
315.     for i in range(len(data_input)):
316.         d_sh = RHOB_VP[0][:, i]
317.         d_sns = RHOB_VP[1][:, i]
318.         d_sn_c1 = RHOB_VP[2][:, i]

```

```

319.         d_sn_gs = RHOB_VP[3][:, i]
320.         d_sn_wt = RHOB_VP[4][:, i]
321.         vp_data = [d_sh, d_snsh, d_sn_cl, d_sn_gs, d_sn_wt]
322.
323.         PlotHistPDF(vp_data, xlabel=label[i], title=data_input[
324.                     i], data_labels=v_data_sets_labels)
325.
326.
327. def PlotZones(well, pos, ax, zones, TOPE, BASE, VSH_CUTOFF):
328.     ''' Plot zones defined by lithology '''
329.     colors = ["green", "blue"]
330.
331.     for key, value in pos.iteritems():
332.         i = key - 1
333.         data = well.data[value]
334.
335.         ax[i].set_xlim(np.nanmin(data), np.nanmax(data))
336.         ax[i].set_ylim(TOPE, BASE)
337.         ax[i].invert_yaxis()
338.         if value == "DT":
339.             ax[i].invert_xaxis()
340.
341.         if value == "VSH":
342.             ax[i].set_xlim(0, 1)
343.             x3 = np.empty(data.shape[0])
344.             x3.fill(VSH_CUTOFF)
345.             ax[i].fill_betweenx(well.data["DEPTH"], data, x3,
346.                                where=data > VSH_CUTOFF, color='gray', alpha=0.5)
347.             ax[i].fill_betweenx(well.data["DEPTH"], data, x3,
348.                                where=data <= VSH_CUTOFF, color='yellow', alpha=0.5)
349.             ax[i].plot(x3, well.data["DEPTH"], 'k-', lw=0.5)
350.
351.         for k in range(2):
352.             for j in range(zones.shape[0]):
353.                 if zones[j][0] == k + 4:
354.                     ax[i].fill_between([np.nanmin(data),
355.                                         np.nanmax(data)],
356.                                         zones[j][1],
357.                                         zones[j][2],
358.                                         color=colors[k],
359.                                         alpha=0.4,
360.                                         linewidth=0.1)
361.
362.
363. def ScatterHist(x, y):
364.     '''
365.     'Lorem
366.     '''
367.
368.     global bins
369.
370.     nullfmt = NullFormatter() # no labels
371.
372.     # definitions for the axes
373.     left, width = 0.1, 0.65
374.     bottom, height = 0.1, 0.65
375.     bottom_h = left_h = left + width + 0.02
376.
377.     rect_scatter = [left, bottom, width, height]
378.     rect_histx = [left, bottom_h, width, 0.2]
379.     rect_histy = [left_h, bottom, 0.2, height]
380.
381.     # start with a rectangular Figure

```

```

382.     plt.figure(1, figsize=(8, 8))
383.
384.     axScatter = plt.axes(rect_scatter)
385.     axHistx = plt.axes(rect_histx)
386.     axHisty = plt.axes(rect_histy)
387.
388.     # no labels
389.     axHistx.xaxis.set_major_formatter(nullfmt)
390.     axHisty.yaxis.set_major_formatter(nullfmt)
391.
392.     # the scatter plot:
393.     axScatter.scatter(x, y)
394.
395.     axScatter.set_xlim((np.min(x), np.max(x)))
396.     axScatter.set_ylim((np.min(y), np.max(y)))
397.
398.     # bins = 30
399.     axHistx.hist(x, bins=bins, normed=True)
400.     axHisty.hist(y, bins=bins, orientation='horizontal', normed=True)
401.
402.     axHistx.set_xlim(axScatter.get_xlim())
403.     axHisty.set_ylim(axScatter.get_ylim())
404.
405.     plt.show()
406.
407.
408. def ContourBI(well, data_input, lit=None, prob=None, createMatrix =False):
409.     '''
410.     Lorem
411.     '''
412.
413.     curves = well.curves.deliver()
414.     if lit is None:
415.         lit = well.data["LIT"]
416.
417.     var = [0, 0]
418.     var[0] = well.data[data_input[0]]
419.
420.     label = [0, 0]
421.
422.     label[0] = data_input[0] + \
423.         " (" + curves[data_input[0]]['units'].lower() + ")"
424.
425.     if len(data_input) == 2:
426.         var[1] = well.data[data_input[1]]
427.         RHOB_VP = SplitByLit(var[0], var[1], lit)
428.         label[1] = data_input[1] + \
429.             " (" + curves[data_input[1]]['units'].lower() + ")"
430.
431.     [H_norm, H2_norm, H_norm_prob, H2_norm_prob] = PlotContourBI(RHOB_VP[0][:, 1],
432. RHOB_VP[0][:, 0],
433. RHOB_VP[1][:, 1], RHOB_VP[1][:, 0],
434. labels=label, legends=["Shale", "Sand"], prob=prob)
435.
436.     # np.savetxt('H_norm.txt', H_norm, delimiter=',')
437.     # np.savetxt('H2_norm.txt', H2_norm, delimiter=',')
438.     # np.savetxt('H_norm_prob.txt', H_norm_prob, delimiter=',')
439.     # np.savetxt('H2_norm_prob.txt', H2_norm_prob, delimiter=',')
440.
441.     if (createMatrix):
442.         # No a-priori

```

```

443.         [matrix1,df1] = CreateMatrixBI(np.array((H_norm, H2_norm)), ["Shale", "Sand"])
444.
445.         #Con A-priori
446.         [matrix2,df2] = CreateMatrixBI(np.array((H_norm_prob, H2_norm_prob)), ["Shale", "Sand"])
447.
448.         return df1, df2
449.
450.     else:
451.         print("Error 506")
452.
453.
454. def PlotContourBI(x, y, x2, y2, title=None, labels=[], legends=[], prob=None):
455.     '''
456.     'Lorem
457.     '''
458.     global bins
459.     # bins = 30
460.
461.     H, xedges, yedges = np.histogram2d(y, x, bins=bins)
462.     H_prob = H * prob['shale']
463.     H2, xedges2, yedges2 = np.histogram2d(y2, x2, bins=bins)
464.     H2_prob = H2 * prob['sand_shale']
465.
466.     H_norm = H / np.nanmax(H)
467.     H2_norm = H2 / np.nanmax(H2)
468.
469.     if np.nanmax(H_prob) >= np.nanmax(H2_prob):
470.         prob_max = np.nanmax(H_prob)
471.     else:
472.         prob_max = np.nanmax(H2_prob)
473.
474.     H_norm_prob = H_prob / prob_max
475.     H2_norm_prob = H2_prob / prob_max
476.
477.     # Contours
478.     X = np.resize(xedges, bins)
479.     Y = np.resize(yedges, bins)
480.
481.     X2 = np.resize(xedges2, bins)
482.     Y2 = np.resize(yedges2, bins)
483.
484.     def set_levels(H, level_min=0.2):
485.         level_max = np.max(H)
486.         level_step = (level_max - level_min + 0.1) * 10
487.         if (level_step) >= 10:
488.             level_step = 10
489.         levels = np.linspace(level_min, level_max, num=level_step)
490.
491.         return levels
492.
493.     levels = set_levels(H_norm)
494.     levels2 = set_levels(H_norm, 0.05)
495.
496.     # levels2 = set_levels(H2_norm)
497.     # levels_h_prob = set_levels(H_prob, prob['shale'])
498.     # levels_h2_prob = set_levels(H2_prob, prob['sand_shale'])
499.
500.     # fig = plt.figure()
501.     # ax = fig.add_subplot(111)
502.     fig, ax = plt.subplots(1, 2, figsize=(10, 5))
503.

```

```

504.     CS1 = ax[0].contour(X, Y, H_norm, levels, cmap=plt.get_cmap("jet"))
505.     CS12 = ax[1].contour(
506.         X, Y, H_norm_prob, levels2, cmap=plt.get_cmap("jet"))
507.
508.     # set levels as previous levels
509.     CS2 = ax[0].contour(X2, Y2, H2_norm,
510.         CS1.levels, cmap=plt.get_cmap("gist_rainbow"),
511.         alpha=0.6)
512.     CS22 = ax[1].contour(X2, Y2, H2_norm_prob,
513.         levels2, cmap=plt.get_cmap("gist_rainbow"),
514.         alpha=0.6)
515.
516.     # print(levels_h2_prob)
517.
518.     # plt.xlabel(labels[0])
519.     # plt.ylabel(labels[1])
520.     # plt.title(labels[0] + " vs " + labels[1])
521.
522.     ax[0].set_xlabel(labels[0])
523.     ax[0].set_ylabel(labels[1])
524.     ax[0].set_title(labels[0] + " vs " + labels[1] + " [No Prob. A-priori]")
525.
526.     lines = [CS1.collections[0], CS2.collections[0]]
527.     ax[0].legend(lines, legends)
528.
529.     ax[1].set_xlabel(labels[0])
530.     ax[1].set_ylabel(labels[1])
531.     ax[1].set_title(labels[0] + " vs " + labels[1] + " [Con Prob. A-priori]")
532.
533.     lines = [CS12.collections[0], CS22.collections[0]]
534.     ax[1].legend(lines, legends)
535.     plt.show()
536.
537.     return H_norm, H2_norm, H_norm_prob, H2_norm_prob
538.     #

```

## Archivo las\_reader.py

```

1.  """LAS File Reader
2.
3.  The main class defined here is LASReader, a class that reads a LAS file
4.  and makes the data available as a Python object.
5.  """
6.
7.  # Copyright (c) 2011, Warren Weckesser
8.  #
9.  # Permission to use, copy, modify, and/or distribute this software for any
10. # purpose with or without fee is hereby granted, provided that the above
11. # copyright notice and this permission notice appear in all copies.
12. #
13. # THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
14. # WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
15. # MERCHANTABILITY AND FITNESS.  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
16. # ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
17. # WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
18. # ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
19. # OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
20.
21. import re

```

```

22. import keyword
23.
24. import numpy as np
25.
26.
27. def isidentifier(s):
28.     if s in keyword.kwlist:
29.         return False
30.     return re.match(r'^[a-z_][a-z0-9_]*$', s, re.I) is not None
31.
32.
33. def _convert_to_value(s):
34.     try:
35.         value = int(s)
36.     except ValueError:
37.         try:
38.             value = float(s)
39.         except ValueError:
40.             value = s
41.     return value
42.
43.
44. class LASError(Exception):
45.     pass
46.
47.
48. class LASItem(object):
49.
50.     """This class is just a namespace, holding the attributes 'name',
51.     'units', 'data', 'value', and 'descr'. 'value' is the numerical
52.     value of 'data', if it has a numerical value (specifically, if
53.     int() or float() don't raise an exception when given the value
54.     of the 'data' attribute).
55.
56.     A class method, from_line(cls, line), is provided to parse
57.     a line from a LAS file and create a LASItem instance.
58.     """
59.
60.     def __init__(self, name, units='', data='', descr=''):
61.         self.name = name
62.         self.units = units
63.         self.data = data
64.         self.value = _convert_to_value(data)
65.         self.descr = descr
66.
67.     def __str__(self):
68.         s = ("name='%s', units='%s', data='%s', descr='%s'" %
69.             (self.name, self.units, self.data, self.descr))
70.         return s
71.
72.     def __repr__(self):
73.         s = str(self)
74.         return "LASItem(%s)" % s
75.
76.     @classmethod
77.     def from_line(cls, line):
78.         first, descr = line.rsplit(':', 1)
79.         descr = descr.strip()
80.         name, mid = first.split('.', 1)
81.         name = name.strip()
82.         if mid.startswith(' '):
83.             # No units
84.             units = ''

```

```

85.         data = mid
86.     else:
87.         units_data = mid.split(None, 1)
88.         if len(units_data) == 1:
89.             units = units_data[0]
90.             data = ''
91.         else:
92.             units, data = units_data
93.         return LASItem(name=name, units=units, data=data.strip(),
94.                        descr=descr.strip())
95.
96.
97. def _read_wrapped_row(f, n):
98.     """Read a "row" of data from the Ascii section of a "wrapped" LAS file.
99.
100.    `f` must be a file object opened for reading.
101.    `n` is the number of fields in the row.
102.
103.    Returns the list of floats read from the file.
104.    """
105.    depth = float(f.readline().strip())
106.    values = [depth]
107.    while len(values) < n:
108.        new_values = [float(s) for s in f.readline().split()]
109.        values.extend(new_values)
110.    return values
111.
112.
113. def _read_wrapped_data(f, dt):
114.     data = []
115.     ncols = len(dt.names)
116.     while True:
117.         try:
118.             row = _read_wrapped_row(f, ncols)
119.         except Exception:
120.             break
121.         data.append(tuple(row))
122.     data = np.array(data, dtype=dt)
123.     return data
124.
125.
126. class LASSection(object):
127.
128.     """Represents a "section" of a LAS file.
129.
130.     A section is basically a collection of items, where each item has the
131.     attributes 'name', 'units', 'data' and 'descr'.
132.
133.     Any item in the section whose name is a valid Python identifier is
134.     also attached to the object as an attribute. For example, if `s` is a
135.     LASSection instance, and the corresponding section in the LAS file
136.     contained this line:
137.
138.     FD      .K/M3                999.9999      : Fluid Density
139.
140.     then the item may be referred to as `s.FD` (in addition to the longer
141.     `s.items['FD']`).
142.
143.     Attributes
144.     -----
145.     items : dict
146.         The keys are the item names, and the values are LASItem instances.
147.     names : list

```

```

148.         List of item names, in the order they were read from the LAS file.
149.
150.         """
151.
152.     def __init__(self):
153.         # Note: In Python 2.7, 'items' could be an OrderedDict, and
154.         # then 'names' would not be necessary--one could use items.keys().
155.         self.items = dict()
156.         self.names = []
157.
158.     def add_item(self, item):
159.         self.items[item.name] = item
160.         self.names.append(item.name)
161.         if isidentifier(item.name) and not hasattr(self, item.name):
162.             setattr(self, item.name, item)
163.
164.     def display(self):
165.         for name in self.names:
166.             item = self.items[name]
167.             namestr = name
168.             if item.units != '':
169.                 namestr = namestr + (" (%s)" % item.units)
170.             print "%-16s %-30s [%s]" % (namestr, "'''''' + item.data + ''",
171.                                     item.descr)
172.
173.     def deliver(self):
174.         output = {}
175.         for name in self.names:
176.             output[name] = {}
177.
178.             item = self.items[name]
179.
180.             output[name]['units'] = item.units.lower()
181.             output[name]['data'] = item.data
182.             output[name]['desc'] = item.descr
183.
184.         return output
185.
186.
187. class LASReader(object):
188.
189.     """The LASReader class holds data from a LAS file.
190.
191.     This reader only handles LAS 2.0 files (as far as I know).
192.
193.     Constructor
194.     -----
195.     LASReader(f, null_subs=None)
196.
197.     f : file object or string
198.         If f is a file object, it must be opened for reading.
199.         If f is a string, it must be the filename of a LAS file.
200.         In that case, the file will be opened and read.
201.
202.     Attributes for LAS Sections
203.     -----
204.     version : LASSection instance
205.         This LASSection holds the items from the '~V' section.
206.
207.     well : LASSection instance
208.         This LASSection holds the items from the '~W' section.
209.
210.     curves : LASSection instance

```

```

211.         This LASSection holds the items from the '~C' section.
212.
213.     parameters : LASSection instance
214.         This LASSection holds the items from the '~P' section.
215.
216.     other : str
217.         Holds the contents of the '~O' section as a single string.
218.
219.     data : numpy 1D structured array
220.         The numerical data from the '~A' section. The data type
221.         of the array is constructed from the items in the '~C'
222.         section.
223.
224.     Other attributes
225.     -----
226.     data2d : numpy 2D array of floats
227.         The numerical data from the '~A' section, as a 2D array.
228.         This is a view of the same data as in the `data` attribute.
229.
230.     wrap : bool
231.         True if the LAS file was wrapped. (More specifically, this
232.         attribute is True if the data field of the item with the
233.         name 'WRAP' in the '~V' section has the value 'YES'.)
234.
235.     vers : str
236.         The LAS version. (More specifically, the value of the data
237.         field of the item with the name 'VERS' in the '~V' section).
238.
239.     null : float or None
240.         The numerical value of the 'NULL' item in the '~W' section.
241.         The value will be None if the 'NULL' item was missing.
242.
243.     null_subs : float or None
244.         The value given in the constructor, to be used as the
245.         replacement value of each occurrence of `null_value` in
246.         the log data. The value will be None (and no substitution
247.         will be done) if the `null_subs` argument is not given to
248.         the constructor.
249.
250.     start : float, or None
251.         Numerical value of the 'STRT' item from the '~W' section.
252.         The value will be None if 'STRT' was not given in the file.
253.
254.     start_units : str
255.         Units of the 'STRT' item from the '~W' section.
256.         The value will be None if 'STRT' was not given in the file.
257.
258.     stop : float
259.         Numerical value of the 'STOP' item from the '~W' section.
260.         The value will be None if 'STOP' was not given in the file.
261.
262.     stop_units : str
263.         Units of the 'STOP' item from the '~W' section.
264.         The value will be None if 'STOP' was not given in the file.
265.
266.     step : float
267.         Numerical value of the 'STEP' item from the '~W' section.
268.         The value will be None if 'STEP' was not given in the file.
269.
270.     step_units : str
271.         Units of the 'STEP' item from the '~W' section.
272.         The value will be None if 'STEP' was not given in the file.
273.

```

```

274.     """
275.
276.     def __init__(self, f, null_subs=None):
277.         """f can be a filename (str) or a file object.
278.
279.         If 'null_subs' is not None, its value replaces any values in the data
280.         that matches the NULL value specified in the Version section of the LAS
281.         file.
282.         """
283.         self.null = None
284.         self.null_subs = null_subs
285.         self.start = None
286.         self.start_units = None
287.         self.stop = None
288.         self.stop_units = None
289.         self.step = None
290.         self.step_units = None
291.
292.         self.version = LASSection()
293.         self.well = LASSection()
294.         self.curves = LASSection()
295.         self.parameters = LASSection()
296.         self.other = ''
297.         self.data = None
298.
299.         self._read_las(f)
300.
301.         self.data2d = self.data.view(float).reshape(-1, len(self.curves.items))
302.         if null_subs is not None:
303.             self.data2d[self.data2d == self.null] = null_subs
304.
305.     def _read_las(self, f):
306.         """Read a LAS file.
307.
308.         Returns a dictionary with keys 'V', 'W', 'C', 'P', 'O' and 'A',
309.         corresponding to the sections of a LAS file. The values associated
310.         with keys 'V', 'W', 'C' and 'P' will be lists of Item instances. The
311.         value associated with the 'O' key is a list of strings. The value
312.         associated with the 'A' key is a numpy structured array containing the
313.         log data. The field names of the array are the mnemonics from the
314.         Curve section of the file.
315.         """
316.         opened_here = False
317.         if isinstance(f, basestring):
318.             opened_here = True
319.             f = open(f, 'r')
320.
321.         self.wrap = False
322.
323.         line = f.readline()
324.         current_section = None
325.         current_section_label = ''
326.         while not line.startswith('~A'):
327.             if not line.startswith('#'):
328.                 if line.startswith('~'):
329.                     if len(line) < 2:
330.                         raise LASError("Missing section character after '~'.")
331.                     current_section_label = line[1:2]
332.                     other = False
333.                     if current_section_label == 'V':
334.                         current_section = self.version
335.                     elif current_section_label == 'W':
336.                         current_section = self.well

```

```

337.         elif current_section_label == 'C':
338.             current_section = self.curves
339.         elif current_section_label == 'P':
340.             current_section = self.parameters
341.         elif current_section_label == 'O':
342.             current_section = self.other
343.             other = True
344.         else:
345.             raise LASError("Unknown section '%s'" % line)
346.     elif current_section is None:
347.         raise LASError("Missing first section.")
348.     else:
349.         if other:
350.             # The 'Other' section is just lines of text, so we
351.             # assemble them into a single string.
352.             self.other += line
353.             current_section = self.other
354.         else:
355.             # Parse the line into a LASItem and add it to the
356.             # current section.
357.             m = LASItem.from_line(line)
358.             current_section.add_item(m)
359.             # Check for the required items whose values we'll
360.             # store as attributes of the LASReader instance.
361.             if current_section == self.version:
362.                 if m.name == 'WRAP':
363.                     if m.data.strip() == 'YES':
364.                         self.wrap = True
365.                 if m.name == 'VERS':
366.                     self.vers = m.data.strip()
367.             if current_section == self.well:
368.                 if m.name == 'NULL':
369.                     self.null = float(m.data)
370.                 elif m.name == 'STRT':
371.                     self.start = float(m.data)
372.                     self.start_units = m.units
373.                 elif m.name == 'STOP':
374.                     self.stop = float(m.data)
375.                     self.stop_units = m.units
376.                 elif m.name == 'STEP':
377.                     self.step = float(m.data)
378.                     self.step_units = m.units
379.             line = f.readline()
380.
381.             # Finished reading the header--all that is left is the numerical
382.             # data that follows the '~A' line. We'll construct a structured
383.             # data type, and, if the data is not wrapped, use numpy.loadtxt
384.             # to read the data into an array. For wrapped rows, we use the
385.             # function _read_wrapped() defined elsewhere in this module.
386.             # The data type is determined by the items from the '~Curves' section.
387.             dt = np.dtype([(name, float) for name in self.curves.names])
388.             if self.wrap:
389.                 a = _read_wrapped_data(f, dt)
390.             else:
391.                 a = np.loadtxt(f, dtype=dt)
392.             self.data = a
393.
394.         if opened_here:
395.             f.close()
396.
397.
398. if __name__ == "__main__":
399.     import sys

```

```

400.
401.     las = LASReader(sys.argv[1], null_subs=np.nan)
402.     print "wrap? ", las.wrap
403.     print "vers? ", las.vers
404.     print "null =", las.null
405.     print "start =", las.start
406.     print "stop  =", las.stop
407.     print "step  =", las.step
408.     print "Version ---"
409.     las.version.display()
410.     print "Well ---"
411.     las.well.display()
412.     print "Curves ---"
413.     las.curves.display()
414.     print "Parameters ---"
415.     las.parameters.display()
416.     print "Other ---"
417.     print las.other
418.     print "Data ---"
419.     print las.data2d

```

Archivo las\_tools.py

```

1.  #!/usr/bin/env python
2.  # -*- coding: utf-8 -*-
3.
4.  """ Lorem ipsum dolor sit amet. """
5.
6.  import os
7.
8.  from math import floor
9.  import numpy as np
10. import numpy.lib.recfunctions as rfn
11. # from pint import UnitRegistry
12. from las_reader import LASItem
13.
14.
15. # # Create the Unit registry.
16. # ureg = UnitRegistry()
17. # Q_ = ureg.Quantity
18. # # Append the new definitions
19. # full_path = os.path.realpath(__file__)
20. # ureg.load_definitions(os.path.dirname(full_path) + '/pint_definitions.txt')
21.
22.
23. def joinArrays(array1, array2):
24.     """ Join 2 structured arrays.
25.         array1: CP['Block']
26.         array2: CP['Reservoir']
27.     """
28.     return rfn.merge_arrays((array1, array2), flatten=True, usemask=False)
29.
30.
31. def remove_field_name(a, name):
32.     """ Remove a column from a structured np.array """
33.     names = list(a.dtype.names)
34.     if name in names:
35.         names.remove(name)
36.     b = a[names]

```

```

37.     return b
38.
39.
40. def LASselect(well, selection):
41.     """ Lorem ipsum dolor sit amet. """
42.     pass
43.
44.
45. # def ChangeUnits(well, logname, toUnits):
46. #     """ Change log unts. """
47.
48. #     curve = well.curves.deliver()
49.
50. #     old = Q_(well.data[logname], curve[logname]["units"])
51. #     new = old.to(toUnits)
52.
53. #     return new.magnitude
54.
55.
56. def AddCurve(well, data, kwargs):
57.     """ Lorem ipsum dolor """
58.     logname = kwargs["name"].upper()
59.
60.     try:
61.         params_data = kwargs["data"].strip()
62.     except KeyError:
63.         params_data = "unknown"
64.
65.     params = LASItem(name=logname,
66.                     units=kwargs["units"],
67.                     data=params_data,
68.                     descr=kwargs["descr"].strip())
69.
70.     # TODO: Check if the new log exists in the curve section
71.     well.curves.add_item(params)
72.
73.     # TODO: Create a Zero array and then replace data
74.     new = np.array(data,
75.                   dtype=[(logname, '<f8')])
76.
77.     try:
78.         well.data = joinArrays(well.data, new)
79.     except ValueError:
80.         purged_data = remove_field_name(well.data, logname)
81.         well.data = joinArrays(purged_data, new)
82.
83.     # return well
84.
85. def FilterData(x, y, top, bottom, ref=0):
86.     xy = np.vstack((y, x)).T
87.
88.     cond = np.logical_and(xy[:, ref] >= top,
89.                          xy[:, ref] <= bottom)
90.
91.     xy = xy[cond]
92.
93.     return xy[:, 1], xy[:, 0]
94.
95.
96. def SetLitZones(LIT, DEPTH):
97.     """ Crea un vector con informaci3n sobre las litologÃas """
98.
99.     zones = np.zeros((LIT.shape[0], 4))

```

```

100.     j = 0
101.     last_lit = 0
102.     num = 1
103.     for i in range(LIT.shape[0] - 1):
104.         type_lit = LIT[i]
105.
106.         if type_lit == last_lit:
107.             zones[j][0] = type_lit
108.             zones[j][2] = DEPTH[i]
109.             num = num + 1
110.         else:
111.             last_lit = type_lit
112.             zones[j][3] = num
113.             num = 1
114.
115.             j = j + 1
116.             zones[j][1] = DEPTH[i]
117.
118.     # Limpiamos la informacion de las zonas
119.     zones = zones[zones[:, 0] > 0]
120.
121.     return zones
122.
123.
124. def SplitByLit(v1, v2, lit):
125.     ''' Split values of v1 and v2 by lithology'''
126.     # Valores de V1 y V2 para la zona impermeable
127.     V1_SHALE = np.zeros(v1.shape[0])
128.     V2_SHALE = np.zeros(v2.shape[0])
129.
130.     # Valores de V1 y V2 para las arenas sucias
131.     V1_SAND_SHALE = np.zeros(v1.shape[0])
132.     V2_SAND_SHALE = np.zeros(v2.shape[0])
133.
134.     # Valores de V1 y V2 para las arenas limpias
135.     V1_SAND_CLEAN = np.zeros(v1.shape[0])
136.     V2_SAND_CLEAN = np.zeros(v2.shape[0])
137.
138.     # Valores de V1 y V2 para las arenas con gas
139.     V1_SAND_GAS = np.zeros(v1.shape[0])
140.     V2_SAND_GAS = np.zeros(v2.shape[0])
141.
142.     # Valores de V1 y V2 para las arenas con gas
143.     V1_SAND_WATER = np.zeros(v1.shape[0])
144.     V2_SAND_WATER = np.zeros(v2.shape[0])
145.
146.     # Valores de V1 y V2 para la zona impermeable
147.     for i in range(0, v1.shape[0]):
148.         if (lit[i] == 1):
149.             V1_SHALE[i] = v1[i]
150.             V2_SHALE[i] = v2[i]
151.         else:
152.             V1_SHALE[i] = np.nan
153.             V2_SHALE[i] = np.nan
154.
155.     # Valores de V1 y V2 para las arenas sucias
156.     for i in range(0, v1.shape[0]):
157.         if (lit[i] == 2):
158.             V1_SAND_SHALE[i] = v1[i]
159.             V2_SAND_SHALE[i] = v2[i]
160.         else:
161.             V1_SAND_SHALE[i] = np.nan
162.             V2_SAND_SHALE[i] = np.nan

```

```

163.
164.     # Valores de V1 y V2 para las arenas limpias
165.     for i in range(0, v1.shape[0]):
166.         if (lit[i] == 3):
167.             V1_SAND_CLEAN[i] = v1[i]
168.             V2_SAND_CLEAN[i] = v2[i]
169.         else:
170.             V1_SAND_CLEAN[i] = np.nan
171.             V2_SAND_CLEAN[i] = np.nan
172.
173.     # Valores de V1 y V2 para las arenas con gas
174.     for i in range(0, v1.shape[0]):
175.         if (lit[i] == 4):
176.             V1_SAND_GAS[i] = v1[i]
177.             V2_SAND_GAS[i] = v2[i]
178.         else:
179.             V1_SAND_GAS[i] = np.nan
180.             V2_SAND_GAS[i] = np.nan
181.
182.     # Valores de V1 y V2 para las arenas con gas
183.     for i in range(0, v1.shape[0]):
184.         if (lit[i] == 5):
185.             V1_SAND_WATER[i] = v1[i]
186.             V2_SAND_WATER[i] = v2[i]
187.         else:
188.             V1_SAND_WATER[i] = np.nan
189.             V2_SAND_WATER[i] = np.nan
190.
191.     for i in range(0, v1.shape[0]):
192.         if (lit[i] == 5):
193.             V1_SAND_WATER[i] = v1[i]
194.             V2_SAND_WATER[i] = v2[i]
195.         else:
196.             V1_SAND_WATER[i] = np.nan
197.             V2_SAND_WATER[i] = np.nan
198.
199.     def clean(array):
200.         var1 = np.zeros(array[0].shape[0])
201.         var2 = np.zeros(array[1].shape[0])
202.         for i in range(0, array[0].shape[0]):
203.             if ~np.isnan(array[0][i]) and ~np.isnan(array[1][i]):
204.                 var1[i] = array[0][i]
205.                 var2[i] = array[1][i]
206.             else:
207.                 var1[i] = np.nan
208.                 var2[i] = np.nan
209.
210.         return np.array([var1[np.isfinite(var1)], var2[np.isfinite(var2)]])
211.
212.     V_SHALE = clean([V1_SHALE, V2_SHALE])
213.     V_SAND_SHALE = clean([V1_SAND_SHALE, V2_SAND_SHALE])
214.     V_SAND_CLEAN = clean([V1_SAND_CLEAN, V2_SAND_CLEAN])
215.     V_SAND_GAS = clean([V1_SAND_GAS, V2_SAND_GAS])
216.     V_SAND_WATER = clean([V1_SAND_WATER, V2_SAND_WATER])
217.
218.     return V_SHALE.T, V_SAND_SHALE.T, V_SAND_CLEAN.T, V_SAND_GAS.T, V_SAND_WATER.T
219.
220.
221. def getMax(a, b):
222.     ''' Return maximum value between two arrays '''
223.     if (np.nanmax(a) > np.nanmax(b)):
224.         max_x = np.nanmax(a)
225.     else:

```

```

226.         max_x = np.nanmax(b)
227.
228.     return max_x
229.
230.
231. def getMin(a, b):
232.     ''' Return minimum value between two arrays '''
233.     if (np.nanmin(a) < np.nanmin(b)):
234.         min_x = np.nanmin(a)
235.     else:
236.         min_x = np.nanmin(b)
237.
238.     return min_x
239.
240.
241. def getValuesZone(well, log, zone):
242.     '''Lorem'''
243.     cond = np.logical_and(well.data["DEPTH"] >= zone[0],
244.                          well.data["DEPTH"] <= zone[1])
245.
246.     return well.data[log][cond]
247.
248.
249. def AreaCurve(x, y, a, b):
250.     '''Lorem'''
251.
252.     a = find_nearest(x,a);
253.     b = find_nearest(x,b);
254.
255.     aIndex = numpy.where(x==a);
256.     bIndex = numpy.where(x==b);
257.
258.     area = 0;
259.     for i in range(aIndex, bIndex):
260.         height = y[i];
261.         base = x[i+1] - x[i] ;
262.         area = area + (base*height);
263.     return area;
264.
265.
266. def find_nearest(array,value):
267.     '''Find nearest value in array'''
268.     idx = (np.abs(array-value)).argmin()
269.     return array[idx]
270.
271.
272.
273. def RollingWindow(a, window):
274.     '''Taken form
275.         http://nbviewer.ipynb.org/github/seg/tutorials/blob/master/1406\_Make\_a\_synthetic/how\_to\_make\_synthetic.ipynb#Fixing-/-editing-the-logs'''
276.     shape = a.shape[:-1] + (a.shape[-1] - window + 1, window)
277.     strides = a.strides + (a.strides[-1],)
278.     rolled = np.lib.stride_tricks.as_strided(a, shape=shape, strides=strides)
279.     return rolled
280.
281.
282. def Despike(curve, curve_sm, max_clip):
283.     '''Taken form
284.         http://nbviewer.ipynb.org/github/seg/tutorials/blob/master/1406\_Make\_a\_synthetic/how\_to\_make\_synthetic.ipynb#Fixing-/-editing-the-logs'''
285.     spikes = np.where(curve - curve_sm > max_clip)[0]
286.     spukes = np.where(curve_sm - curve > max_clip)[0]

```

```

287.     out = np.copy(curve)
288.     out[spikes] = curve_sm[spikes] + max_clip # Clip at the max allowed diff
289.     out[spukes] = curve_sm[spukes] - max_clip # Clip at the min allowed diff
290.     return out
291.
292. def SmothData(input_data, window):
293.     .....
294.     data_out = np.median(RollingWindow(input_data,window), -1)
295.     data_out = np.pad(data_out, int(floor(window/2)), mode='edge')
296.
297.     return data_out

```

## Apéndice F. Cuaderno IPython de ejemplo para la interpretación petrofísica

El siguiente código muestra un cuaderno IPython, convertido a código Python para facilitar su lectura. Este código representa el código genérico para la realización de la interpretación petrofísica.

```

1. # coding: utf-8
2.
3. ## Análisis petrofísico del un Pozo de ejemplo
4.
5. # In[1]:
6.
7. from __future__ import absolute_import, print_function, division
8.
9. from IPython.external import mathjax; #mathjax.install_mathjax()
10. from IPython.core.display import Image
11. from IPython.display import display
12. import numpy as np
13. import matplotlib as mpl
14. import matplotlib.pyplot as plt
15. from scipy.stats import itemfreq
16.
17. import libs as ls
18.
19. well_dir = u"../pozos-hr/pozos/"
20. img_dir = u"../pozos-notebooks/img/"
21.
22. # soporte de gráficos dentro del cuaderno
23. get_ipython().magic(u'matplotlib inline')
24. get_ipython().magic(u'pylab inline --no-import-all')
25. # Aumenta el tamaño de las imágenes mostradas
26. matplotlib.rcParams['savefig.dpi'] = 2 * matplotlib.rcParams['savefig.dpi']
27.
28. get_ipython().magic(u'load_ext autoreload')
29. get_ipython().magic(u'autoreload 2')
30.
31.
32. # In[2]:
33.
34. well_name = "nombre_del_pozo"
35.

```

```

36.
37. # In[3]:
38.
39. well_dir + well_name + "_logs.LAS"
40.
41.
42. # In[4]:
43.
44. well = ls.LASReader( well_dir + well_name + "_logs.las", null_subs=np.nan)
45. c = {}
46. c_ltd = {}
47.
48.
49. ### Registros disponibles
50.
51. # In[5]:
52.
53. well.curves.display()
54.
55.
56. ### Información previa:
57.
58. # In[7]:
59.
60. EXTRA_DEPTH = 20
61. TOPE = 1550 - EXTRA_DEPTH
62. BASE = 1691 + EXTRA_DEPTH
63.
64.
65. ### Constantes
66.
67. # In[8]:
68.
69. # Vshale de corte
70. VSH_CUTOFF = 0.4
71. # Factor de cementación
72. a = 0.62
73. # Exponente de cementación
74. m = 2.15
75. #
76. n = 2.0
77.
78.
79. ### Suavizado de Curvas
80.
81. # In[9]:
82.
83. window = 11 # the length of filter is 13 samples or ~ 1.5 metres
84.
85. #####
86. c["GRS"] = ls.SmothData(well.data["GR"],window)
87. params = dict(name='GRS',
88.               units='API',
89.               descr='Smoth GR'
90.               )
91.
92. ls.AddCurve(well, c["GRS"], params)
93.
94. #####
95. c["PHINS"] = ls.SmothData(well.data["PHIN"],window)
96. c["PHINS"] = np.clip(c["PHINS"] , 0, 1)
97. params = dict(name='PHINS',
98.               units='fraction',

```

```

99.         descr='Smothered PHIN'
100.     )
101.
102. ls.AddCurve(well, c["PHINS"], params)
103.
104. #####
105. c["PHIDS"] = ls.SmothData(well.data["PHID"],window)
106. c["PHIDS"] = np.clip(c["PHIDS"] , 0, 1)
107. params = dict(name='PHIDS',
108.              units='fraction',
109.              descr='Smothered PHID'
110.              )
111.
112. ls.AddCurve(well, c["PHIDS"], params)
113.
114. #####
115. c["RHOB"] = ls.SmothData(well.data["RHOB"],window)
116. params = dict(name='RHOB',
117.              units='kg/m3',
118.              descr='Smothered RHOB'
119.              )
120.
121. ls.AddCurve(well, c["RHOB"], params)
122.
123. #####
124. c["DTPS"] = ls.SmothData(well.data["DT"],window)
125. params = dict(name='DTPS',
126.              units='us/m',
127.              descr='Smothered DT'
128.              )
129.
130. ls.AddCurve(well, c["DTPS"], params)
131.
132. #####
133. c["DTSS"] = ls.SmothData(well.data["DTs"],window)
134. params = dict(name='DTSS',
135.              units='us/m',
136.              descr='Smothered DTs'
137.              )
138.
139. ls.AddCurve(well, c["DTSS"], params)
140.
141.
142. # In[10]:
143.
144. fig, ax = ls.LASPlot(well,["GR", "GRS"], zone = [TOPE, BASE])
145. fig, ax = ls.LASPlot(well,["DT", "DTPS"], zone = [TOPE, BASE])
146. fig, ax = ls.LASPlot(well,["DTs", "DTSS"], zone = [TOPE, BASE])
147. fig, ax = ls.LASPlot(well,["RHOB", "RHOB"], zone = [TOPE, BASE])
148.
149.
150. ## Cálculo del VShale
151.
152. ### Cálculo de Igr
153.
154. # In[11]:
155.
156. xy = np.vstack((well.data['DEPTH'], well.data['GRS'])).T
157.
158. cond = np.logical_and(xy[:, 0] >= TOPE,
159.                      xy[:, 0] <= BASE)
160. xy = xy[cond]
161.

```

```

162. # Obtengo los valores de GRcn y GRsh
163. GRCN = np.nanmin(xy[:, 1])
164. GRSH = np.nanmax(xy[:, 1])
165. GRSH = 125
166.
167. print("GRCn = ", GRCN)
168. print("GRsh = ", GRSH)
169.
170. # Calculo de índice de Gamma Ray
171. c["IGR"] = (well.data["GRS"] - GRCN) / (GRSH - GRCN)
172.
173.
174. ##### Cálculo de volume de arcilla:
175.
176. # In[12]:
177.
178. c["VSH"] = 0.33 * ((2 ** (2 * c["IGR"])) - 1)
179. c["VSH"] = np.clip(c["VSH"], 0, 1)
180.
181. params = dict(name='VSH',
182.               units='fraction',
183.               descr='Shale Volume'
184.               )
185.
186. ls.AddCurve(well, c["VSH"], params)
187.
188.
189. # In[13]:
190.
191. fig, ax = ls.LASPlot(well, ["VSH"], zone = [TOPE, BASE])
192. x, y = ls.FilterData(well.data["VSH"], well.data["DEPTH"], TOPE, BASE)
193.
194. # Linea vertical
195. x2 = np.empty(x.shape[0])
196. x2.fill(VSH_CUTOFF)
197.
198. vlines(VSH_CUTOFF, TOPE, BASE, color='k', lw=0.5)
199.
200. # # Llenamos las zonas permeables e impermeables
201. wh1 = x > VSH_CUTOFF
202. wh2 = x <= VSH_CUTOFF
203. ax.fill_betweenx(y, x, x2, where=wh1, color='red', alpha=0.5)
204. ax.fill_betweenx(y, x, x2, where=wh2, color='yellow', alpha=0.5)
205.
206.
207. ## Creación de la curva de litología
208.
209. # La **Zona impermeable** tendrá un valor de **1**
210. # La **zona permeable sucia** tendrá un valor de **2**
211. # La **zona permeable limpia** tendrá un valor de **3**
212.
213. # In[14]:
214.
215. # Creamos la Curva de Litología
216. c["LIT_SIMPLE"] = np.empty(well.data.shape[0])
217. i = 0
218. for v in well.data["VSH"]:
219.     if ~np.isnan(v):
220.         if v > VSH_CUTOFF:
221.             c["LIT_SIMPLE"][i] = 1 # Shale
222.         elif v <= VSH_CUTOFF:
223.             c["LIT_SIMPLE"][i] = 2 # Sand Shale
224.     else:

```

```

225.         c["LIT_SIMPLE"][i] = np.nan
226.     i += 1
227.
228.
229.     params = dict(name='LIT_SIMPLE',
230.                  units='NONE',
231.                  descr='Lithology (Shale & Sand)'
232.                  )
233.
234.     ls.AddCurve(well, c["LIT_SIMPLE"], params)
235.
236.
237. # In[15]:
238.
239. # Creamos la Curva de Litología
240. c["LIT"] = np.empty(well.data.shape[0])
241. i = 0
242. for v in well.data["VSH"]:
243.     if ~np.isnan(v):
244.         if v > VSH_CUTOFF:
245.             c["LIT"][i] = 1 # Shale
246.         elif v >= (VSH_CUTOFF-0.1) and v <= VSH_CUTOFF:
247.             c["LIT"][i] = 2 # Sand Shale
248.         else:
249.             c["LIT"][i] = 3 # Sand Clean
250.     else:
251.         c["LIT"][i] = np.nan
252.     i += 1
253.
254.
255. ##### Cálculo de Saturación de Agua y Resistividad de Agua Aparente
256.
257. ##### Porosidad Promedio
258.
259. # In[16]:
260.
261. # Calculamos la presencia de gas
262. c["PAYGAS"] = np.empty(well.data.shape[0])
263.
264. for i in range(0, well.data.shape[0] - 1):
265.     PHIN = well.data["PHINS"][i]
266.     PHID = well.data["PHIDS"][i]
267.     if ~(np.isnan(PHIN) or np.isnan(PHID)):
268.         if (PHIN < PHID) and abs(PHIN - PHID) >= 0.07:
269.             c["PAYGAS"][i] = 1 # Gas
270.         else:
271.             c["PAYGAS"][i] = 0 # Not gas
272.     else:
273.         c["PAYGAS"][i] = np.nan
274.
275.
276. # Actualización de la curva de litología, para **arena limpias con gas** tomamos un v
    alor de *4*
277.
278. # In[17]:
279.
280. # Actualizamos la curva de litología
281. for i in range(0, well.data.shape[0] - 1):
282.     if (c["LIT"][i] == 3) and (c["PAYGAS"][i] == 1):
283.         c["LIT"][i] = 4 # Arena limpia con GAS
284.         #print(well.data["DEPTH"][i])
285.
286.

```

```

287. # In[18]:
288.
289. # Calculamos la porosidad promedio
290. c["PHIA"] = np.empty(well.data.shape[0])
291.
292. for i in range(0, well.data.shape[0] - 1):
293.     PHIN = well.data["PHINS"][i]
294.     PHID = well.data["PHIDS"][i]
295.     if c["LIT"][i] == 4:
296.         c["PHIA"][i] = sqrt(((PHIN ** 2) + (PHID ** 2)) / 2)
297.     else:
298.         c["PHIA"][i] = (PHIN + PHID) / 2
299.
300.
301.
302. params = dict(name='PHIA',
303.               units='fraction',
304.               descr='Appearant Porosity'
305.               )
306.
307. ls.AddCurve(well, c["PHIA"], params)
308.
309.
310. # In[19]:
311.
312. fig, ax = ls.LASPlot(well, ["PHIDS"], zone = [TOPE, BASE])
313.
314.
315. ax.set_title("Porosidad vs Profundidad")
316. ax.legend(loc='upper center', bbox_to_anchor=(-0.3, 0.3),
317.          fancybox=True, ncol=1, fontsize=10)
318. ax.plot(well.data["PHINS"], well.data["DEPTH"], 'b', lw=0.5, label="PHINS")
319.
320. ax.set_xlim(-0.15, 0.45)
321. # Añadimos la leyenda
322. ax.legend(loc='upper center', bbox_to_anchor=(-0.3, 0.3),
323.          fancybox=True, ncol=1, fontsize=10)
324.
325.
326. ##### Resistividad de agua parente
327.
328. # Está dada por:
329. # # $$ R_{wA} = \frac{R_t \phi_A^m}{a} $$
330.
331. # In[20]:
332.
333. # Calculo de Rwa
334. c["RWA"] = (well.data["ILD"] * (c["PHIA"] ** m)) / a
335.
336.
337. ##### Saturación de Agua
338.
339. # Está dada por:
340. # # $$ S_{wA} = \frac{a R_{wA}}{R_t \phi_A^m} ^{1/n} $$
341.
342. # Una vez recalculado el Vshl, se buscará el **menor valor de Rwa**, dentro de las **zonas permeables**
343. # con un **valor bajo de Vshl e ILD** (*acuifero*), el cual deberá corresponder al **valor de Rwa** (*Manual de PetroLuke, 2004, Mendoza*)
344.
345. # In[21]:
346.
347. # Linea vertical

```

```

348. x2 = np.empty(well.data["VSH"].shape[0])
349. x2.fill(VSH_CUTOFF)
350.
351. x3 = np.empty(well.data["VSH"].shape[0])
352. x3.fill(VSH_CUTOFF - 0.1)
353.
354. # Creamos los gráficos
355. fig, ax = ls.LASPlot(well,["VSH", "ILD"], zone = [TOPE, BASE])
356. ax1 = ax[0]
357. ax2 = ax[1]
358.
359. ax2.plot(x2, well.data["DEPTH"], 'k-', lw=0.5) # vlines(0.4, TOPE, BASE)
360. ax2.plot(x3, well.data["DEPTH"], 'k-', lw=0.5) # vlines(0.4, TOPE, BASE)
361.
362. ax2.set_xlim(0, 1)
363.
364. # Llenamos las zonas permeables e impermeables
365. ax2.fill_betweenx(well.data["DEPTH"], well.data["VSH"], x2, where=well.data["VSH"] >
366.                 VSH_CUTOFF, color='red', alpha=0.5)
367. ax2.fill_betweenx(well.data["DEPTH"], well.data["VSH"], x2,
368.                 where=well.data["VSH"] <= VSH_CUTOFF, color='yellow', alpha=0.5)
369.
370.
371.
372. # Una vez recalculado el Vsh1, se buscará el **menor valor de Rwa**, dentro de las **
zonas permeables**
373. # con un **valor bajo de Vsh1 e ILD** (*acuífero*), el cual deberá corresponder al *
*valor de Rw** (*Manual de PetroLuke, 2004, Mendoza*)
374.
375. # In[22]:
376.
377. #
378. c_ltd["ILD"] = well.data["ILD"] < 7
379. c_ltd["VSH"] = well.data["VSH"] < 0.15
380. for i in range(0, well.data.shape[0] - 1):
381.     if c_ltd["ILD"][i] and c_ltd["VSH"][i]:
382.         print(well.data["ILD"][i], well.data["VSH"][i], c["RWA"][i])
383.
384.
385. # El valor de Rw a escoger es ** 0.345231252054 omh-m**
386.
387. # In[23]:
388.
389. # Cálculo saturación de agua aparente
390. Rw = 0.345231252054
391.
392. c["SWA"] = np.empty(well.data.shape[0])
393.
394. for i in range(0, c["SWA"].shape[0] - 1):
395.     # SWA[i] = (a*Rw) / (ILD[i]* (PHIA[i]**m))**
396.     c["SWA"][i] = ( (a * Rw) / (well.data["ILD"][i] * (c["RWA"][i] ** m)) ) ** (1.0 /
m)
397.
398.
399. # In[24]:
400.
401. # Smoth and clip values
402. c["SWA"] = np.clip(c["SWA"], 0, 1)
403. c["SWA"] = ls.SmothData(c["SWA"],window)
404.
405.
406. # In[25]:
407.

```

```

408. fig, ax = ls.LASPlotCustom(c["SWA"], well.data["DEPTH"], r'$Sw_a$',
409.                             'Profundidad (m)', r'$Sw_a$ vs Profundidad', 'red')
410. ax.set_ylim(TOPE, BASE)
411. ax.set_xlim(0, 1)
412. ax.invert_yaxis()
413.
414.
415. # In[26]:
416.
417. # Actualizamos la curva de litología
418. num_lit_5 = 0
419. for i in range(0, len(c["LIT"]) - 1):
420.     if (c["LIT"][i] == 3) and (c["SWA"][i] >= 1):
421.         c["LIT"][i] = 5 # Arena limpia con Agua
422.         num_lit_5 = num_lit_5 + 1
423.         #print(well.data["DEPTH"][i])
424.
425. print(num_lit_5)
426.
427.
428. # In[27]:
429.
430. p = {}
431. p["name"] = "LIT"
432. p["units"] = "none"
433. p["descr"] = "Lithology"
434.
435. ls.AddCurve(well, c["LIT"], p)
436.
437.
438. ##### Creación del vector con las zonas según su litología
439.
440. # In[28]:
441.
442. zones = ls.SetLitZones(well.data["LIT"], well.data["DEPTH"])
443.
444.
445. #### Sónico, Densidad, Gamma Ray y Resistividad
446.
447. # In[29]:
448.
449. curves = ["GRS", "VSH", "ILD", "DTPS", "RHOB"]
450. fig, ax, pos = ls.LASPlot( well, curves, zone=[TOPE, BASE], returnpos=True, sharey = T
    rue)
451.
452. ls.PlotZones(well, pos, ax, zones, TOPE, BASE, VSH_CUTOFF)
453.
454.
455. # In[30]:
456.
457. curves = ["PHINS", "PHIDS", "PHIA"]
458. fig, axes, pos = ls.LASPlot( well, curves, zone=[TOPE, BASE], returnpos=True, sharey =
    True)
459.
460. ls.PlotZones(well, pos, axes, zones, TOPE, BASE, VSH_CUTOFF)
461.
462.
463. ## Probabilidades a priori - Litología
464.
465. # In[31]:
466.
467. c2 = {}
468. c2["LIT_SIMPLE"] = ls.getValuesZone(well, "LIT_SIMPLE", zone = [TOPE, BASE])

```

```

469. c2["LIT"] = ls.getValuesZone(well, "LIT", zone = [TOPE, BASE])
470.
471. c["LIT_SIMPLE"].shape[0], c2["LIT_SIMPLE"].shape[0]
472.
473.
474. # In[32]:
475.
476. c["LIT"].shape[0], c2["LIT"].shape[0]
477.
478.
479. # In[ ]:
480.
481. #itemfreq(c2["LIT_SIMPLE"]), itemfreq(c2["LIT"])
482.
483.
484. # In[ ]:
485.
486. def get_prob_freq(log, operation):
487.     return log.compress(( eval(operation) ).flat).shape[0] / log.shape[0]
488.
489. prob_simple = {}
490. prob_simple["shale"] = get_prob_freq(c2["LIT_SIMPLE"], "log==1")
491. prob_simple["sand_shale"] = get_prob_freq(c2["LIT_SIMPLE"], "log==2")
492. prob_simple["nan"] = get_prob_freq(c2["LIT_SIMPLE"], "np.isnan(log)")
493.
494. prob_simple
495.
496.
497. # In[35]:
498.
499. prob = {}
500. prob["shale"] = get_prob_freq(c2["LIT"], "log==1")
501. prob["sand_shale"] = get_prob_freq(c2["LIT"], "log==2")
502. prob["sand_clean"] = get_prob_freq(c2["LIT"], "log==3")
503. prob["sand_gas"] = get_prob_freq(c2["LIT"], "log==4")
504. prob["sand_water"] = get_prob_freq(c2["LIT"], "log==5")
505. prob["nan"] = get_prob_freq(c2["LIT"], "np.isnan(log)")
506.
507. prob
508.
509.
510. # In[36]:
511.
512. a = ['shale', 'sand_shale', 'sand_clean', 'sand_gas', 'sand_water']
513.
514.
515. ## Curvas estadísticas asociadas a diferenciación de fluidos
516.
517. # ## Gráficos de Vs y Vp
518. #
519.
520. # In[37]:
521.
522. c["VP"] = (1. / well.data["DTPS"]) * 1e6 # m/s
523. c["VS"] = (1. / well.data["DTSS"]) * 1e6 # m/s
524.
525.
526. # In[38]:
527.
528. ls.PlotPair(well, ["DTSS", "DTPS"], prob=prob)
529.
530.
531. ### Impedancia acústica y elástica

```

```

532.
533. # La *impedancia acústica* está dada por:
534. #
535. # ## $$ Z = \rho * V_p $$
536.
537. # In[39]:
538.
539. c["RHOB"] = well.data['RHOB'] * 0.001 # gr/cc
540. c["AI"] = c["VP"] * c["RHOB"] # m/s * gr/cc
541.
542. params = {}
543. params["name"] = "AI"
544. params["units"] = "m/s * gr/cc"
545. params["data"] = ""
546. params["descr"] = "Acoustic Impedance"
547.
548. ls.AddCurve(well, c["AI"], params)
549.
550.
551. # La *impedancia alástica* está dada por:
552. # # $$ K = \frac{1}{2} ( \frac{V^2_{s_2}}{V^2_{p_2}} + \frac{V^2_{s_1}}{V^2_{p_1}} ) $$
553. #
554. # # $$ EI = [ V_p^{(1+\tan^2\theta)} V_s^{-8K\sin^2\theta} \rho^{(1-4K\sin^2\theta)} ] $$
555.
556. # In[40]:
557.
558. # Calculo de la impedancia elástica
559. row_num = well.data.shape[0]
560. c["EI"] = np.zeros(row_num)
561. c["EI"][row_num - 1] = np.nan
562. theta = 30
563. theta_rad = np.pi * theta / 180
564.
565.
566. for i in range(row_num - 1):
567.     k1 = (c["VS"][i + 1] ** 2) / (c["VP"][i + 1] ** 2)
568.     k2 = (c["VS"][i] ** 2) / (c["VP"][i] ** 2)
569.     k = 0.5*(k1 + k2)
570.
571.
572.     aa = c["VP"][i] ** (1 + np.tan(theta_rad) ** 2)
573.     bb = c["VS"][i] ** (-8 * k * np.sin(theta_rad) ** 2)
574.     cc = c["RHOB"][i] ** (1 - 4 * k * np.sin(theta_rad) ** 2)
575.
576.     c["EI"][i] = aa * bb * cc
577.
578. params = dict(name='EI',
579.               units='m/s * gr/cc',
580.               descr='Elastic Impedance'
581.               )
582.
583. ls.AddCurve(well, c["EI"], params)
584.
585.
586. # In[41]:
587.
588. ls.PlotPair(well, ["AI", "EI"], prob = prob)
589.
590.
591. ### Calculo de parámetros elásticos
592.

```

```

593. # ### Módulo de Cizalla
594. #
595. # # $$ \mu = \rho * V_s^{2} $$
596.
597. # In[42]:
598.
599. # units = kg/m3 * m/s => kg/(m2*s)
600. c["MU"] = well.data["RHOB"] * (c["VS"] ** 2)
601.
602.
603. # In[43]:
604.
605. params = dict(name='cizalla',
606.               units='kg/(m2*s)',
607.               descr='Módulo de Cizalla'
608.               )
609.
610. ls.AddCurve(well, c["MU"], params)
611.
612.
613. # ### Constante de Lamé
614. #
615. # # $$ \lambda = \rho ( V_p^{2} - 2 V_s^{2} ) $$
616.
617. # In[44]:
618.
619. # units = kg/m3 * m2/s2 => kg/(m*s2)
620. c["LABMDA"] = well.data["RHOB"] * ( (c["VP"] ** 2) - 2*(c["VS"] ** 2) )
621.
622.
623. # In[45]:
624.
625. params = dict(name='lame',
626.               units='kg/(m*s2)',
627.               descr='Constante Lamé'
628.               )
629.
630. ls.AddCurve(well, c["LABMDA"], params)
631.
632.
633. # ### Relación de Poisson
634. #
635. # # $$ \nu = \frac{1}{2} ( \frac{V_p^2 - 2 V_s^2}{V_p^2 - V_s^2} ) $$
636.
637. # In[46]:
638.
639. nu_a = (c["VP"] ** 2) - 2*(c["VS"] ** 2)
640. nu_b = (c["VP"] ** 2) - (c["VS"] ** 2)
641.
642. c["NU"] = 0.5 * nu_a / nu_b # units: none
643.
644.
645. # In[47]:
646.
647. params = {}
648. params["name"] = "poisson"
649. params["units"] = "none"
650. params["data"] = ""
651. params["descr"] = "Relacion Poisson"
652.
653. ls.AddCurve(well, c["NU"], params)
654.
655.

```

```

656. # ### Módulo de Young
657. #
658. # $$ E = \frac{\rho V^2_{\{s\}} ( 3 V^2_{\{p\}} - 4 V^2_{\{s\}} ) }{ V^2_{\{p\}} - V^2_{\{s\}} } $$
659. #
660.
661. # In[48]:
662.
663. e_a = well.data["RHOB"] * c["VS"] ** 2
664. e_b = (3 * c["VP"] **2) - 4 * c["VS"] ** 2
665. e_c = (c["VP"] ** 2) - (c["VS"] ** 2)
666.
667. # units: kg/m3 * m2/s2 => kg/(m*s2)
668. c["E"] = (e_a * e_b) / e_c
669.
670.
671. # In[49]:
672.
673. params = {}
674. params["name"] = "young"
675. params["units"] = "kg/(m2 * s)"
676. params["data"] = ""
677. params["descr"] = "Modulo Young"
678.
679. ls.AddCurve(well, c["E"], params)
680.
681.
682. ##### Gráficas
683.
684. # In[50]:
685.
686. curves = ["LAME", "POISSON", "YOUNG", "CIZALLA"]
687. fig, ax, pos = ls.LASPlot( well, curves, zone=[TOPE, BASE], returnpos=True, sharey = T
rue)
688.
689. ls.PlotZones(well, pos, ax, zones, TOPE, BASE, VSH_CUTOFF)
690.
691.
692. # In[51]:
693.
694. ls.PlotPair(well, ["POISSON", "YOUNG"], prob =prob)
695.
696.
697. # In[52]:
698.
699. ls.PlotPair(well, ["LAME", "CIZALLA"], prob =prob)
700.
701.
702. ## Curvas estadísticas asociadas a diferenciación de litología
703.
704. ##### Vp y Vs
705.
706. # In[53]:
707.
708. [matrix_no_apriori, matrix_apriori] = ls.PlotPair(well, ["DTSS", "DTPS"], lit = well.d
ata["LIT_SIMPLE"], prob =prob_simple, createMatrix =True)
709.
710.
711. # In[54]:
712.
713. print("----- Dts -----\n")
714. display(matrix_no_apriori[0]), display(matrix_apriori[0])
715. print("----- Dt -----\n")
716. display(matrix_no_apriori[1]),display( matrix_apriori[1])

```

```

717.
718.
719. ##### Impedancia acústica y elástica
720.
721. # In[55]:
722.
723. [matrix_no_apriori, matrix_apriori] = ls.PlotPair(well, ["AI", "EI"], lit = well.data[
    "LIT_SIMPLE"], prob = prob_simple, createMatrix = True)
724.
725.
726. # In[56]:
727.
728. print("----- AI -----\\n")
729. display(matrix_no_apriori[0]), display(matrix_apriori[0])
730. print("----- EI -----\\n")
731. display(matrix_no_apriori[1]), display( matrix_apriori[1])
732.
733.
734. ##### Parámetros elásticos
735.
736. # In[57]:
737.
738. [matrix_no_apriori, matrix_apriori] = ls.PlotPair(well, ["POISSON", "YOUNG"], lit = we
    ll.data["LIT_SIMPLE"], prob = prob, createMatrix = True)
739.
740.
741. # In[58]:
742.
743. print("----- Poisson -----\\n")
744. display(matrix_no_apriori[0]), display(matrix_apriori[0])
745. print("----- Young -----\\n")
746. display(matrix_no_apriori[1]), display( matrix_apriori[1])
747.
748.
749. # In[59]:
750.
751. [matrix_no_apriori, matrix_apriori] = ls.PlotPair(well, ["LAME", "CIZALLA"], lit=well.
    data["LIT_SIMPLE"], prob=prob, createMatrix = True)
752.
753.
754. # In[60]:
755.
756. print("----- Lame -----\\n")
757. display(matrix_no_apriori[0]), display(matrix_apriori[0])
758. print("----- Shear -----\\n")
759. display(matrix_no_apriori[1]), display( matrix_apriori[1])
760.
761.
762. ## Gráficos Bivariados
763.
764. # In[ ]:
765.
766. from itertools import combinations
767.
768. curves2 = ["DTSS", "DTPS", "AI", "EI", "RHOB", "LAME", "POISSON", "YOUNG", "CIZALLA"]
769.
770. def longestWord(letters):
771.     letters = [curve + "-" for curve in letters]
772.     result = [''.join(i) for i in combinations(letters, r=2)]
773.     result2 = [c[:-1].split("-") for c in result]
774.     return result2
775.

```

```

776. combinations = longestWord(queries)
777. combinations
778.
779.
780. # In[ ]:
781.
782. for curves_plot in combinations:
783.     print(curves_plot)
784.     [df1, df2] = ls.ContourBI(well, curves_plot, lit = well.data["LIT_SIMPLE"], prob=p
rob_simple, createMatrix =True)
785.     display(df1), display(df2)
786.     print("--")
787.
788.
789. # In[ ]:
790.
791. header = 'DEPTH(m)\t\tLIT_SIMPLE\t\tAcoustic_Impedance(m/s * gr/cc)'
792.
793. a = np.vstack((well.data["DEPTH"], well.data["LIT_SIMPLE"], c["AI"])).T
794.
795. np.savetxt(well_name + '_results.txt', a, fmt=['%.2f', '%g', '%0.4f'], header=header)

```